

An introduction to partial lambda algebras

J.R.B. Cockett and P.J.W. Hofstra*

Department of Computer Science, University of Calgary,
Calgary, T2N 1N4, Alberta, Canada

October 17, 2006

*Partially supported by NSERC, Canada.

Contents

1	Introduction	3
2	Cartesian restriction categories and objects of partial maps	6
2.1	Preliminaries on restriction categories	6
2.2	Cartesian closedness for restriction categories	7
2.3	Exponents in \mathcal{M} -categories	9
2.4	Cartesian closed restriction categories	11
2.5	Notes and references	15
3	Partial combinatory logic and the partial λ-calculus	16
3.1	Partial lambda calculus	16
3.2	Partial combinatory logic	19
3.3	Lambda abstraction and combinatory completeness	20
3.4	Relation between CL^R and the partial λ -calculus	22
3.5	Notes and references	23
4	Combinatory structures in a restriction category	25
4.1	Combinatory Algebras	25
4.2	Turing Objects	27
4.3	Combinatory algebras as models	29
4.4	Partial lambda algebras	29
4.5	Notes and references	31
5	Reflexive objects in a restriction category	33
5.1	Reflexive objects	33
5.2	Interpretation of partial lambda terms	34
5.3	From partial lambda algebras to reflexive objects	38
5.4	Notes and references	45
6	Examples	46
6.1	Directed Complete Partial Orders	46
6.2	The D_∞ -construction	47
6.3	Graph algebras and trees	48
6.4	Notes and references	50
7	Appendix	51

1 Introduction

The purpose of this paper is to give an exposition of the theory of partial combinatory algebras, models of the partial lambda calculus and various related concepts from the point of view of restriction categories. We feel that restriction categories not only simplify and clarify the presentation of the subject matter, but also provide us with precisely the right type of logic to reason about the structures involved.

One of the problems that one encounters when delving into the literature on the subject is the lack of uniformity: there are various different logics for reasoning about partiality, such as the logic of partial terms (LPT) and the logic of partial elements (LPE). Similarly, there are various partial versions of the lambda calculus. These logics are all related (for an excruciatingly detailed account, see Moggi's thesis), but the problem remains that in generalizing a classical result to the partial world, one has to choose versions of the partial analogues of the concepts involved, and these choices will often be governed by practical considerations or personal taste rather than by a methodological principle. In addition, most of the category-theoretic work that has been done takes place in the setting of partial map categories, which are technically not the most convenient setting to work in.

By taking the notion of a restriction category as fundamental we overcome these issues; restriction categories (like any class of categories) come equipped with a term logic, and this term logic will be the logic we use for reasoning about partial structures. For example, the partial version of combinatory logic we use is nothing but (an instantiation of) the term logic for cartesian restriction categories. By adhering to this viewpoint, we obtain a clearer view of the connections among the various concepts, and, in some cases, easier and/or more perspicuous proofs.

But let us first describe the topic of this paper in a bit more detail. Classically, there are tight connections between the following concepts:

- Combinatory Logic
- Combinatory Algebras
- Lambda Calculus
- Lambda Algebras
- Turing Categories
- Reflexive objects in CCC's (cartesian closed categories)

First of all, Combinatory Logic (CL) is an equational theory which aims to capture a notion of abstract computation. Even though this is not immediately clear from the axioms, the essential feature of the theory is that one can mimic lambda abstraction and do some elementary recursion theory. The models of this theory are precisely the Combinatory Algebras (CAs) which are, loosely

speaking, sets equipped with an application map with the additional property of combinatory completeness. This means that every algebraic (“polynomial”) function is representable by an element.

The lambda calculus is essentially stronger than combinatory logic: there is a sound translation from CL into the lambda calculus, but the translation going the other way (which uses the fact that lambda abstraction can be mimicked in CL) is not sound. This gap is bridged by adding axioms to CL, which makes the theories equivalent. It is possible to do this using only a finite set of closed equations, which, in the literature, go by the name A_β . Now lambda algebras are the models of the lambda calculus; in particular, a lambda algebra is a CA in which some extra axioms hold.

Both classes of models have interesting connections with category theory. A Turing category is a category in which one has a special object (called the Turing object), which has the following two properties: every object is a retract of the Turing object, and the Turing object has an application $T \times T \xrightarrow{\bullet} T$ such that every map in the category is representable by an element of the Turing object. The application map is sometimes called a Turing morphism, or a Kleene-universal morphism. The collection of global elements of a Turing object is a combinatory algebra, and every combinatory algebra can be realized as a Turing object in a certain Turing category. For lambda algebras, more is true: a reflexive object in a cartesian closed category is an object A for which the exponential A^A is a retract of A . The Scott-Koymans theorem states that every such reflexive object gives rise to a lambda algebra, and that every lambda algebra gives rise to a reflexive object. The construction of this reflexive object is done by first forming a monoid of representable endofunctions, and then formally splitting the idempotents in that monoid. The resulting category is a cartesian closed Turing category with a reflexive Turing object.

Given the important role that partiality plays in various branches of computer science and logic, it comes as no surprise that researchers have generalized some of the above concepts and connections to the partial setting. Just to mention a few (more references will be given after each section): Partial Combinatory Algebras (PCAs) have been studied as models of combinatory logic, as a subject in itself (see, for example Inge Bethke’s thesis [Bet88]) and as the key ingredient for realizability models [Lon94, Hof03]. Partial versions of combinatory logic were defined by Beeson, and Scott. In his Ph. D. thesis, Moggi introduced various formal frameworks for dealing with partiality and he compared these frameworks with the existing ones from a proof-theoretic perspective. He did the same for the various extensions of the lambda calculus. A generalization of the Scott-Koymans theorem was announced by R. Pino Pérez and C. Even [Eve95], although a complete proof has not been published. The same authors have obtained a finite axiomatization of (a version of) the partial lambda calculus over (a version of) partial combinatory logic.

As stated before, our main aim in this expository paper is to provide a uniform account of the above results. Aside from a clean presentation of these results in a very general setting, we also obtain some improvements and strengthenings,

and fill in some gaps in the literature. In the notes following each section, we have indicated the origins of the results to the best of our knowledge.

It is our hope, that this material will be an accessible exposition to people with a background in basic category theory; in particular, the notion of a cartesian closed category will be used. Knowledge of the lambda calculus and combinatory logic is a definite advantage, but the presentation is essentially self-contained. We give most of the proofs in reasonable if not full detail; even when the proof seems to be a straightforward adaptation from the total case, there are always subtleties involved because of the partiality, and often these subtleties form the key to understanding the matter. For the easy and straightforward results which are not new, we give brief sketches, leaving the details as an exercise to the interested and/or sceptical reader.

2 Cartesian restriction categories and objects of partial maps

In this section we introduce the categorical machinery necessary for the results. First, we rehearse some standard facts about restriction categories. Then we develop the theory of (partial) exponentials in a cartesian restriction category. In the literature, partial function spaces in categories of partial maps have been considered and have led to the notion of a partial cartesian closed category. Here, we present the notion of a cartesian closed restriction category, which coincides with the former notion in the case when all idempotents split.

2.1 Preliminaries on restriction categories

We start by giving a brief recapitulation of cartesian restriction categories. A **restriction category** is a category \mathbf{C} endowed with a combinator $\overline{(-)}$, sending $f : A \rightarrow B$ to $\overline{f} : A \rightarrow A$, such that the following axioms are satisfied:

$$\begin{aligned}
 R.1 \quad & f\overline{f} = f \\
 R.2 \quad & \overline{f\overline{g}} = \overline{g}\overline{f} \quad \text{whenever } \text{dom}(f) = \text{dom}(g) \\
 R.3 \quad & \overline{g\overline{f}} = \overline{g\overline{g}} \quad \text{whenever } \text{dom}(f) = \text{dom}(g) \\
 R.4 \quad & \overline{g}\overline{f} = \overline{f}\overline{g}\overline{f} \quad \text{whenever } \text{cod}(f) = \text{dom}(g)
 \end{aligned}$$

Maps of the form \overline{f} are idempotents, and maps f such that $f = \overline{f}$ are called **restriction idempotents**. The restriction idempotent associated to a map f is to be thought of as (the idempotent capturing) the domain of f . A map f is called **total** if $\overline{f} = 1$. It easily follows from the axioms that every monomorphism is total. Moreover, the total maps form a subcategory of \mathbf{C} , which we will call $\text{Tot}(\mathbf{C})$.

One can also show that restriction categories are locally ordered; for maps $f, g : A \rightarrow B$, say that $f \leq g$ if $f = g\overline{f}$. Informally, this means that the graph of f is included in the graph of g .

An important example is Par , the category of sets and partial maps. More generally, whenever a category \mathbf{D} has a stable system of monics \mathcal{M} , one can form the partial map category $\text{Par}(\mathbf{D}, \mathcal{M})$, and this is a restriction category. Here, a system of monics is said to be stable when it contains all isomorphisms, is closed under composition and when the pullback of each monic in \mathcal{M} (exists and) is again in \mathcal{M} . The partial map category $\text{Par}(\mathbf{D}, \mathcal{M})$ has the same objects as \mathbf{D} ,

but maps from A to B are now represented by a span $A \xleftarrow{m} C \xrightarrow{f} B$, where $m \in \mathcal{M}$ and where f is arbitrary. Two such spans (m, f) and (m', f') are equivalent if there is an isomorphism g such that $m'g = m, f'g = f$. Composition is given by pullback. $\text{Par}(\mathbf{D}, \mathcal{M})$ is a restriction category. A restriction category is of this form when all the idempotents split. In fact, this condition characterizes partial map categories amongst all restriction categories.

It is fairly common practice to take a restriction category \mathbf{C} and formally split a class E of idempotents. The resulting restriction category is then denoted

$K_E(\mathbf{C})$; it has as objects pairs (X, e) where X is an object of \mathbf{C} and e is an idempotent on X in the class E . Its morphisms $f : (X, e) \rightarrow (X', e')$ are maps $f : X \rightarrow X'$ of \mathbf{C} for which $e'fe = f$. The restriction of such a map is defined to be $\bar{f}e$ (or, equivalently, one can take $e'\bar{f}$). Of particular interest is the case where one splits all restriction idempotents of \mathbf{C} ; we will use the notation $K_r(\mathbf{C})$ for that case. A restriction category in which all restriction idempotents split is called **effective**.

An object \top in a restriction category \mathbf{C} is said to be a **restriction terminal object** if for each object A there is a unique total map $!_A : A \rightarrow \top$, such that $!_{\top} = 1_{\top}$, and for each $f : A \rightarrow B$ we have $!_B f = !_A \bar{f}$, as in the diagram below.

$$\begin{array}{ccc} A & \xrightarrow{!_A} & \top \\ f \downarrow & \nearrow & \\ B & & \end{array}$$

A **partial product** of two objects A, B is an object $A \otimes B$ equipped with total projections $\pi_A : A \otimes B \rightarrow A$ and $\pi_B : A \otimes B \rightarrow B$, such that for each C and each pair of maps $f : C \rightarrow A, g : C \rightarrow B$, there is a unique map $\langle f, g \rangle : C \rightarrow A \otimes B$ with the properties that $\pi_A \langle f, g \rangle \leq f, \pi_B \langle f, g \rangle \leq g$ and $\langle f, g \rangle = \bar{f}g$.

$$\begin{array}{ccccc} & & C & & \\ & f \swarrow & \downarrow \langle f, g \rangle & \searrow g & \\ A & \xleftarrow{\pi_A} & A \otimes B & \xrightarrow{\pi_B} & B \end{array}$$

A restriction category is said to be a **cartesian restriction category** if it has a restriction terminal object and for every A, B a partial product $A \otimes B$.

Partial products in a restriction category correspond to genuine products in the total map category. That is, \mathbf{C} is a cartesian restriction category if and only if the category $\mathbf{Tot}(\mathbf{C})$ has finite products.

A functor $F : \mathbf{C} \rightarrow \mathbf{D}$ is called a **restriction functor** if it commutes with the restrictions, i.e. if $F(\bar{f}) = \overline{F(f)}$. A restriction functor is called **cartesian** when it preserves partial products and the restriction terminal object. An important example for us is the global sections functor: when \mathbf{C} has a restriction terminal object \top , this is defined as $\Gamma_t : \mathbf{C} \rightarrow \mathbf{Par}$, where $\Gamma_t(C) = \{f : \top \rightarrow C \mid \bar{f} = 1\}$. In the case that \mathbf{C} is cartesian, it follows that the restriction functor Γ_t preserves the cartesian structure.

2.2 Cartesian closedness for restriction categories

We now investigate the notion of exponent (“space of partial mappings”) in the context of a cartesian restriction category. It turns out that there really are two notions, which coincide when restriction idempotents split.

Definition 2.1 (Restricted Exponential). Let A and B be two objects of \mathbf{C} . A **restricted exponential** for A, B is an object $[A \multimap B]$ together with a map $\epsilon_{A,B} : A \otimes [A \multimap B] \rightarrow B$, such that the following universal property holds:

Given a map $f : A \otimes X \rightarrow B$ and a restriction idempotent $e = \bar{e} : X \rightarrow X$ satisfying $f \circ (1_A \otimes e) = f$, there exists a unique map $\mathbf{tr}(f, e) : X \rightarrow [A \multimap B]$ such that

$$\begin{array}{ccc} A \otimes [A \multimap B] & \xrightarrow{\epsilon_{A,B}} & B \\ \uparrow 1_A \otimes \mathbf{tr}(f, e) & \nearrow f & \\ A \otimes X & & \end{array}$$

commutes, and $\overline{\mathbf{tr}(f, e)} = e$.

We refer to the map $\mathbf{tr}(f, e)$ as the transpose of f (relative to e).

An important special case is where $B = \top$, the tensor unit. In that case, the definition reduces to that of a partial map classifier for A . We can also consider the case where $A = \top$, which amounts to a power object for B . Here, of course, “partial map classifier” and “power object” are to be taken in the appropriate restriction category-theoretic sense.

If we remove the dependence of $\mathbf{tr}(f, e)$ on the idempotent e in the definition we arrive at the usual definition of an exponential for a partial map category.

Definition 2.2 (Exponential). Let A, B be objects of \mathbf{C} . An **exponential** for A, B is an object $[A \multimap B]$ together with a map $\varepsilon_{A,B} : A \otimes [A \multimap B] \rightarrow B$ such that for every map $f : A \otimes X \rightarrow B$ there is a unique total map $\mathbf{tr}(f) : X \rightarrow [A \multimap B]$ for which $f = \varepsilon_{A,B} \circ (1_A \otimes \mathbf{tr}(f))$.

The definition of a restricted exponential is such, that it becomes precisely an exponential when we split restriction idempotents:

Lemma 2.3. Let A, B be objects of \mathbf{C} . An object $[A \multimap B]$ is a restricted exponential in \mathbf{C} if and only if it is an exponential in $K_r(\mathbf{C})$.

Proof. Consider an object (X, e) in $K_r(\mathbf{C})$, where e is a restriction idempotent on X . A map $f : A \otimes (X, e) \rightarrow B$ is precisely a map $A \otimes X \rightarrow B$ for which $f \circ (1 \otimes e) = f$.

Thus we have a one-one correspondence between pairs $(f, e = \bar{e} : X \rightarrow X)$ in \mathbf{C} for which $f \circ (1_A \otimes e) = f$ and maps $f : A \otimes (X, e) \rightarrow B$ in $K_r(\mathbf{C})$.

Given a pair (f, e) , a map $\mathbf{tr}(f, e) : X \rightarrow [A \multimap B]$ (which lives in \mathbf{C}), such that $\epsilon_{A,B} \circ (1_A \otimes \mathbf{tr}(f, e)) = f : A \otimes X \rightarrow B$ and $\overline{\mathbf{tr}(f, e)} = e$ is also a map in $K_r(\mathbf{C})$, namely $\mathbf{tr}(f, e) : (X, e) \rightarrow [A \multimap B]$. To see this, simply calculate $\mathbf{tr}(f, e) \circ e = \mathbf{tr}(f, e) \overline{\mathbf{tr}(f, e)} = \mathbf{tr}(f, e)$. Since $\overline{\mathbf{tr}(f, e)} = e$ is the identity on (X, e) , the map $\mathbf{tr}(f, e)$ is total.

Thus, a morphism $\mathbf{tr}(f, e) : X \rightarrow [A \multimap B]$ for which the equalities $\epsilon_{A,B} \circ (1_A \otimes \mathbf{tr}(f, e)) = f : A \otimes X \rightarrow B$ and $\overline{\mathbf{tr}(f, e)} = e$ hold, is precisely a morphism

$\mathbf{tr}(f, e) : (X, e) \rightarrow [A \multimap B]$ in $K_r(\mathbf{C})$ making the triangle

$$\begin{array}{ccc} A \otimes [A \multimap B] & \xrightarrow{\epsilon_{A,B}} & B \\ \uparrow 1_A \otimes \mathbf{tr}(f, e) & \nearrow f & \\ A \otimes (X, e) & & \end{array}$$

commutative.

This shows that $\epsilon_{A,B} : A \otimes [A \multimap B] \rightarrow B$ is a restricted exponential in \mathbf{C} if and only if it is an exponential in $K_r(\mathbf{C})$. \square

The following corollary is immediate from the previous lemma.

Corollary 2.4. *Let \mathbf{C} be an effective restriction category. Then an object $[A \multimap B]$ is an exponential if and only if it is a restricted exponential.*

Proof. For an effective restriction category \mathbf{C} , we have $\mathbf{C} \simeq K_r(\mathbf{C})$. Now apply the lemma.

Alternatively, one can use the following direct proof. Let $e : X \rightarrow X$ be a restriction idempotent with splitting $e = nm$, where $m : X' \rightarrow X$. For any map $f : A \otimes X \rightarrow B$ with $f(1 \otimes e) = f$, consider $f' = f(1 \otimes m) : A \otimes X' \rightarrow B$. Take the transpose $\mathbf{tr}(f') : X' \rightarrow [A \multimap B]$ of f' . It is now easily verified that the composite $\mathbf{tr}(f') \circ n : X \rightarrow [A \multimap B]$ is the transpose of f with respect to the idempotent e . \square

We can strengthen Lemma 2.3 by replacing the class of restriction idempotents by an arbitrary class of idempotents which contains the class of restriction idempotents. To see this, note first that in an effective restriction category, every idempotent e has a decomposition $e = me'n$, where e' is a total idempotent and where $mn = \bar{e}$. Reasoning as in the proof of the previous lemma, we can find the transpose of a map f with respect to an idempotent e as the composite $\mathbf{tr}(f') \circ n$, where $f' = f(1 \otimes m)$.

We summarize this in the following proposition.

Proposition 2.5. *An object $[A \multimap B]$ is a restricted exponent in \mathbf{C} if and only if it is a restricted exponent in $K_E(\mathbf{C})$ for any class of idempotents E containing the restriction idempotents.*

2.3 Exponents in \mathcal{M} -categories

Whenever one has a category \mathbf{D} with a stable system of monics \mathcal{M} , one can form the associated partial map category $\mathbf{Par}(\mathbf{D}, \mathcal{M})$. Our next task is to see what structure in \mathbf{D} corresponds to having exponents in the partial map category. As we would hope, this is precisely the notion of an object of partial maps.

Definition 2.6 (Object of Partial Maps). *Let \mathbf{D} be a category with finite limits equipped with a stable system of monics \mathcal{M} , and let A, B be objects of \mathbf{D} .*

Then an **object of partial maps** (relative to \mathcal{M}) for A and B is an object $[A \multimap B]$ and a partial map, represented by a span,

$$A \times [A \multimap B] \xleftarrow{\eta_{A,B}} D_{A,B} \xrightarrow{ev_{A,B}} B$$

subject to the requirement that for any partial map $A \times X \xleftarrow{m} U \xrightarrow{f} B$ there exist unique maps $(m, f)_0 : X \rightarrow [A \multimap B]$ and $(m, f)_1 : U \rightarrow D_{A,B}$, such that in the diagram below the triangle commutes and the square is a pullback.

$$\begin{array}{ccc} & & B \\ & \nearrow f & \uparrow ev_{A,B} \\ U & \xrightarrow{(m,f)_1} & D_{A,B} \\ \downarrow m & & \downarrow \eta_{A,B} \\ A \times X & \xrightarrow{1_A \times (m,f)_0} & A \times [A \multimap B] \end{array}$$

Intuitively, the object $D_{A,B}$ is the set $\{(a, f) | f(a) \downarrow\}$. In this definition, all partial maps are, of course, partial maps relative to \mathcal{M} , but we are going to stop mentioning that everywhere.

Lemma 2.7. *Let A, B be objects in a partial map category $\text{Par}(\mathcal{C}, \mathcal{M})$, and write $D = \text{Tot}(\text{Par}(\mathcal{C}, \mathcal{M}))$. Then $[A \multimap B]$ is an exponential in $\text{Par}(\mathcal{C}, \mathcal{M})$ if and only if it is an object of partial maps in D .*

Proof. Consider a span $A \times [A \multimap B] \xleftarrow{\eta_{A,B}} D_{A,B} \xrightarrow{ev_{A,B}} B$. This is the same as a map $\varepsilon_{A,B} : A \times [A \multimap B] \rightarrow B$ in the partial map category.

For a fixed span (m, f) as in the definition, commutative diagrams in $\text{Par}(\mathcal{C}, \mathcal{M})$ of the form

$$\begin{array}{ccc} A \otimes [A \multimap B] & \xrightarrow{\varepsilon_{A,B}} & B \\ \uparrow 1_A \otimes (m,f)_0 & \nearrow f & \\ A \otimes X & & \end{array}$$

correspond to diagrams in D of the form

$$\begin{array}{ccccc} & & U & & \\ & \swarrow m & & \searrow (m,f)_1 & \\ & A \times X & & D_{A,B} & \\ \swarrow 1 & \searrow 1_A \otimes (m,f)_0 & \swarrow \eta_{A,B} & \searrow ev_{A,B} & \\ A \times X & & A \times [A \multimap B] & & B \end{array}$$

where the square is a pullback and $ev_{A,B} \circ (m, f)_1 = f$. \square

Combining this with Proposition 2.5, we get:

Proposition 2.8. *Let A, B be objects of a cartesian restriction category \mathbf{C} . Then $[A \multimap B]$ is a restricted exponential in \mathbf{C} if and only if it is an object of partial maps in $\text{Tot}(K_E(\mathbf{C}))$ for any class of idempotents E which contains the restriction idempotents.*

2.4 Cartesian closed restriction categories

Next, we look at functoriality. We work in the cartesian restriction category \mathbf{C} and assume that for all objects A, B the restricted exponential $[A \multimap B]$ exists. (We assume that we have chosen and fixed, for each A, B , the structure $\epsilon_{A,B} : A \otimes [A \multimap B] \rightarrow B$.) We will summarize this situation by calling \mathbf{C} a ccrc, a cartesian closed restriction category.

Suppose maps $\alpha : A' \rightarrow A$ and $\beta : B \rightarrow B'$ are given. Consider the diagram

$$\begin{array}{ccc} A \otimes [A \multimap B] & \xrightarrow{1_A \otimes [\alpha \multimap B]} & A \otimes [A \multimap B] \\ \alpha \otimes 1 \downarrow & & \downarrow \epsilon_{A,B} \\ A' \otimes [A' \multimap B] & \xrightarrow{\epsilon_{A',B}} & B \end{array}$$

which shows that there is a unique total map $[\alpha \multimap B]$ making the assignment $A \mapsto [A \multimap B]$ into a contravariant functor $[- \multimap B] : \mathbf{C}^{op} \rightarrow \text{Tot}(\mathbf{C})$.

Similarly, the diagram

$$\begin{array}{ccc} A \otimes [A \multimap B] & \xrightarrow{1_A \otimes [A \multimap \beta]} & A \otimes [A \multimap B'] \\ \epsilon_{A,B} \downarrow & & \downarrow \epsilon_{A,B'} \\ B & \xrightarrow{\beta} & B' \end{array}$$

shows that $A \mapsto [A \multimap B]$ is a covariant functor $\mathbf{C} \rightarrow \text{Tot}(\mathbf{C})$.

Combining the two, we have a bifunctor

$$[- \multimap -] : \mathbf{C}^{op} \times \mathbf{C} \rightarrow \text{Tot}(\mathbf{C}).$$

We could also have defined the two components in one step, namely by means of the diagram

$$\begin{array}{ccc} A' \otimes [A \multimap B] & \xrightarrow{1_{A'} \otimes [\alpha \multimap \beta]} & A' \otimes [A' \multimap B'] \\ \alpha \otimes 1 \downarrow & & \downarrow \epsilon_{A',B'} \\ A \otimes [A \multimap B] & \xrightarrow{\epsilon_{A,B}} B \xrightarrow{\beta} & B' \end{array}$$

It is easily verified that this gives the same result, i.e.

$$[\alpha \multimap B] \circ [A \multimap \beta] = [\alpha \multimap \beta] = [A \multimap \beta] \circ [\alpha \multimap B].$$

We are now going to prove that if \mathbf{C} is a ccrc, then so is $K_E(\mathbf{C})$, under certain conditions on the class E . First, we will prove a few lemmas:

Lemma 2.9. *If e_0 is any idempotent on X , then we have $1 \otimes [e_0 \rightarrow Y] = (e_0 \otimes 1) \circ (1 \otimes [e_0 \rightarrow Y]) = (1 \otimes [e_0 \rightarrow Y]) \circ (e_0 \otimes 1)$.*

Proof. Observe that both squares are commutative in the diagram below:

$$\begin{array}{ccccc}
X \otimes [X \rightarrow Y] & \xrightarrow{e_0 \otimes 1} & X \otimes [X \rightarrow Y] & \xrightarrow{1 \otimes [e_0 \rightarrow Y]} & X \otimes [X \rightarrow Y] \\
e_0 \otimes 1 \downarrow & & e_0 \otimes 1 \downarrow & & \downarrow \epsilon_{X,Y} \\
X \otimes [X \rightarrow Y] & \xrightarrow{1} & X \otimes [X \rightarrow Y] & \xrightarrow{\epsilon_{X,Y}} & Y
\end{array}$$

Therefore, the whole diagram commutes and by unicity of the top map we have $(1 \otimes [e_0 \rightarrow Y]) \circ (e_0 \otimes 1) = 1 \otimes [e_0 \rightarrow Y]$. The other identities are obvious now. \square

Lemma 2.10. *Let e_0 again be any idempotent on X , and let e_1 be any idempotent on Y . Then we have*

$$(e_0 \otimes 1) \circ (1 \otimes [e_0 \rightarrow e_1]) = 1 \otimes [e_0 \rightarrow e_1] = (1 \otimes [e_0 \rightarrow e_1]) \circ (e_0 \otimes 1).$$

Proof. We have the following equalities:

$$\begin{aligned}
(e_0 \otimes 1) \circ (1 \otimes [e_0 \rightarrow e_1]) &= (e_0 \otimes 1) \circ (1 \otimes [e_0 \rightarrow Y]) \circ (1 \otimes [X \rightarrow e_1]) \\
&= (1 \otimes [e_0 \rightarrow Y]) \circ (1 \otimes [X \rightarrow e_1]) \\
&= (1 \otimes [e_0 \rightarrow e_1]) \\
&= (1 \otimes [X \rightarrow e_1]) \circ (1 \otimes [e_0 \rightarrow Y]) \\
&= (1 \otimes [X \rightarrow e_1]) \circ (1 \otimes [e_0 \rightarrow Y]) \circ (e_0 \otimes 1) \\
&= (1 \otimes [e_0 \rightarrow e_1]) \circ (e_0 \otimes 1).
\end{aligned}$$

\square

Now we are ready for the construction of general exponentials in $K_E(\mathbb{C})$.

Proposition 2.11. *Let \mathbb{C} be a ccr and let E be a class of idempotents which contains all restriction idempotents and is closed under taking exponents. Then all exponentials exist in $K_E(\mathbb{C})$.*

Proof. Take (X, e_0) and (Y, e_1) where e_0 is a restriction idempotent on X and e_1 is a restriction idempotent on Y . We are going to show that the object $([X \rightarrow Y], [e_0 \rightarrow e_1])$ is the desired exponential. By assumption on the class E , the idempotent $[e_0 \rightarrow e_1]$ is in E , so that the object $([X \rightarrow Y], [e_0 \rightarrow e_1])$ is indeed an object of $K_E(\mathbb{C})$.

Define the evaluation map to be the composite

$$X \otimes [X \rightarrow Y] \xrightarrow{e_0 \otimes 1} X \otimes [X \rightarrow Y] \xrightarrow{\epsilon_{X,Y}} Y \xrightarrow{e_1} Y.$$

We will denote this map by $EV = EV_{e_0, e_1}$. Let us first verify that EV is indeed a map in the category $K_r(\mathcal{C})$. That is, we calculate

$$\begin{aligned}
e_1 \circ EV \circ (e_0 \otimes [e_0 \rightarrow e_1]) &= e_1 \circ (e_1 \circ \epsilon_{X,Y} \circ (e_0 \otimes 1)) \circ (e_0 \otimes [e_0 \rightarrow e_1]) \\
&= e_1 \circ \epsilon_{X,Y} \circ (1 \otimes [e_0 \rightarrow e_1]) \circ (e_0 \otimes 1) \\
&= e_1 \circ \epsilon_{X,Y} \circ (e_0 \otimes 1) \\
&= EV.
\end{aligned}$$

Here, the second equality is a repeated application of Lemma 2.10, and the third equality holds because of the commutativity of the following diagram:

$$\begin{array}{ccccc}
X \otimes [X \rightarrow Y] & \xrightarrow{e_0 \otimes 1} & X \otimes [X \rightarrow Y] & \xrightarrow{1 \otimes [e_0 \rightarrow e_1]} & X \otimes [X \rightarrow Y] \\
& \searrow^{e_0 \otimes 1} & \downarrow^{e_0 \otimes 1} & & \downarrow^{\epsilon_{X,Y}} \\
& & X \otimes [X \rightarrow Y] & \xrightarrow{\epsilon_{X,Y}} & Y \\
& & & \searrow^{e_1} & \downarrow^{e_1} \\
& & & & Y
\end{array}$$

Next, we have to verify the universal property of EV . It is sufficient to show that it holds for objects (Z, e_2) , where e_2 is a restriction idempotent on Z , because we work in an effective restriction category. So, let $f : (X, e_0) \otimes (Z, e_2) \rightarrow (Y, e_1)$ be any map, that is, $e_1 \circ f \circ (e_0 \otimes e_2) = f$. We have to find a unique total map $\mathbf{tr}(f) : (Z, e_2) \rightarrow ([X \rightarrow Y], [e_0 \rightarrow e_1])$ such that the triangle

$$\begin{array}{ccc}
(X, e_0) \otimes ([X \rightarrow Y], [e_0 \rightarrow e_1]) & \xrightarrow{EV} & (Y, e_1) \\
\uparrow^{1 \otimes \mathbf{tr}(f)} & \nearrow f & \\
(X, e_0) \otimes (Z, e_2) & &
\end{array}$$

is commutative.

Since $f = f \circ (1 \otimes e_2)$, we let $\mathbf{tr}(f)$ be the map $\mathbf{tr}(f, e_2)$ obtained from the diagram

$$\begin{array}{ccc}
X \otimes [X \rightarrow Y] & \xrightarrow{\epsilon_{X,Y}} & Y \\
\uparrow^{1 \otimes \mathbf{tr}(f, e_2)} & \nearrow f & \\
X \otimes Z & & \\
\uparrow^{1 \otimes e_2} & & \\
X \otimes Z & &
\end{array}$$

That is, we have equalities $\overline{\mathbf{tr}(f)} = \overline{\mathbf{tr}(f, e_2)} = e_2$ and $f = \epsilon_{X,Y} \circ (1 \otimes \mathbf{tr}(f))$.

We first need to verify that the map $\mathbf{tr}(f)$ obtained in this way is indeed a map $(Z, e_2) \rightarrow ([X \rightarrow Y], [e_0 \rightarrow e_1])$. We have $\mathbf{tr}(f) \circ e_2 = \mathbf{tr}(f)$, so it remains

to be seen that $[e_0 \dashv e_1] \circ \mathbf{tr}(f) = \mathbf{tr}(f)$. Note first that since $e_1 \circ f = f$, we obtain

$$e_1 \circ \epsilon_{X,Y} \circ (1 \otimes \mathbf{tr}(f)) = \epsilon_{X,Y} \circ (1 \otimes \mathbf{tr}(f)). \quad (1)$$

To prove that $[e_0 \dashv e_1] \circ \mathbf{tr}(f) = \mathbf{tr}(f)$, it suffices to see that the diagram

$$\begin{array}{ccc} X \otimes Z & \xrightarrow{1 \otimes \mathbf{tr}(f)} & X \otimes [X \dashv Y] \\ \downarrow 1 \otimes \mathbf{tr}(f) & & \downarrow \epsilon_{X,Y} \\ X \otimes [X \dashv Y] & & \\ \downarrow e_0 \otimes 1 & & \\ X \otimes [X \dashv Y] & \xrightarrow{\epsilon_{X,Y}} & Y \xrightarrow{e_1} Y \end{array}$$

commutes: for clearly replacing the top arrow with $(1 \otimes [e_0 \dashv e_1]) \circ (1 \otimes \mathbf{tr}(f))$ makes the diagram commute, and by uniqueness of transposes it will follow that $\mathbf{tr}(f) = [e_0 \dashv e_1] \circ \mathbf{tr}(f)$. But the equality $f \circ (e_0 \otimes 1) = f$ implies $\epsilon_{X,Y} \circ (e_0 \otimes \mathbf{tr}(f)) = \epsilon_{X,Y} \circ (1 \otimes \mathbf{tr}(f))$; combine this with Equation 1 and we get

$$\epsilon_{X,Y} \circ (e_0 \otimes \mathbf{tr}(f)) = e_1 \circ \epsilon_{X,Y} \circ (1 \otimes \mathbf{tr}(f))$$

which says exactly that the diagram commutes.

Next, note that $\mathbf{tr}(f)$ is total (as a map $(Z, e_2) \rightarrow ([X \dashv Y], [e_0 \dashv e_1])$ in $K_r(\mathbf{C})$), simply because the restriction of $\mathbf{tr}(f)$ in $K_r(\mathbf{C})$ is constructed as $\mathbf{tr}(f) \circ e_2 = e_2$.

Then we calculate:

$$\begin{aligned} EV \circ (1 \otimes \mathbf{tr}(f)) &= e_1 \circ \epsilon_{X,Y} \circ (e_0 \otimes 1) \circ (1 \otimes \mathbf{tr}(f)) \\ &= e_1 \circ \epsilon_{X,Y} \circ (1 \otimes \mathbf{tr}(f)) \\ &= \epsilon_{X,Y} \circ (1 \otimes \mathbf{tr}(f)) \\ &= f. \end{aligned}$$

which shows that $\mathbf{tr}(f)$ indeed makes the relevant triangle commute.

Finally, we have to prove that $\mathbf{tr}(f)$ is unique with this property. But this is simply seen by tracing back along the above equations; if a map $g : X \otimes Z \rightarrow Y$ satisfies the equations

$$[e_0 \dashv e_1] \circ g \circ (e_0 \otimes e_2) = g, \quad EV \circ (1 \otimes g) = f,$$

then we first derive $1 \otimes g = (1 \otimes g) \circ (e_0 \otimes 1)$. On the other hand, we get $[X \dashv e_1] \circ g = g$, from which we get the equation $e_1 \circ \epsilon_{X,Y} \circ (1 \otimes g) = \epsilon_{X,Y} \circ (1 \otimes g)$. This means that $\epsilon_{X,Y} \circ (1 \otimes g) = f$ in \mathbf{C} , so that $\mathbf{tr}(f) = g$. This completes the proof. \square

Corollary 2.12. *For a cartesian restriction category \mathbf{C} and a class of idempotents E which contains all restriction idempotents and is closed under taking exponents, the following statements are equivalent:*

1. \mathcal{C} is a cartesian closed restriction category;
2. $\text{Tot}(K_E(\mathcal{C}))$ has all objects of partial maps.

2.5 Notes and references

The basic theory of restriction categories is described in detail in [Coc02], and limits in restriction categories have been studied in [Coc04]. The notion of an exponential in a category of partial maps has been developed by Curien and Obtulowicz in their paper [Cur87], where the authors obtain a characterization of toposes in terms of their associated partial map categories. Along these lines, we should also mention the recent paper by Schröder [Sch04], where various properties of partial cartesian closed categories are investigated. In the context of restriction categories, no complete account of exponentials has been presented previously, although the material in [Coc03] contains the development of partial map classifiers in restriction categories..

3 Partial combinatory logic and the partial λ -calculus

Classically, combinatory algebras are precisely models of a certain equational theory, called combinatory logic (CL for short). There are translations from CL to the lambda calculus and back, but CL is essentially weaker than the lambda calculus: CL can be soundly interpreted in the lambda calculus, but not vice versa. The theories can be made equivalent by adding five axioms to CL, which in the literature go by the name A_β . A combinatory algebra satisfying these extra axioms admits a sound interpretation of the lambda calculus and is therefore called a lambda algebra.

In this section, we present analogues in the partial world of combinatory logic and the lambda calculus. We define translations between partial CL and the partial lambda calculus and indicate (a proof is in the appendix) that the partial lambda calculus can be axiomatized over partial CL by means of a finite set of closed axioms.

3.1 Partial lambda calculus

We define a partial variant of the classical lambda calculus, which differs from the ordinary lambda calculus in that terms can be restricted to other terms. First, we take care of term formation. Throughout, we fix a countably infinite set $Var = \{x_0, x_1, \dots\}$ of variables.

Definition 3.1 (Partial lambda terms). *The set of partial lambda terms is denoted Λ^R and is inductively defined as follows:*

- $Var \subseteq \Lambda^R$
- $M, N \in \Lambda^R \implies (MN) \in \Lambda^R$
- $M \in \Lambda^R, x \in Var \implies (\lambda x.M) \in \Lambda^R$
- $M, D \in \Lambda^R \implies M|_D \in \Lambda^R$

Clearly, the first three formation rules are exactly those of the classical lambda calculus. The fourth rule says that if we have a term M and another term D , then we can form $M|_D$, which we think of as M restricted to the domain of D . We don't distinguish notationally between restrictions consisting of one term, consisting of a sequence of terms, or a finite set of terms; the restriction operation is assumed to be associating on the left, so by $M|_{D_1, \dots, D_n}$ we mean $(\dots(M|_{D_1})|_{D_2} \dots)|_{D_n}$.

We adopt the standard conventions of avoiding variable clashes and of identifying terms modulo alpha-conversion. The usual notion of substitution is extended with the clause

$$M|_D[x := N] = M[x := N]|_{D[x := N]}$$

L.1	$M = M$	
L.2	$M = N \implies N = M$	
L.3	$M = N, N = P \implies M = P$	
L.4	$M = N \implies MZ = NZ$	
L.5	$M = N \implies ZM = ZN$	
L.6	$M _M = M$	
L.7	$M _{(N _D)} = M _{D,N}$	
L.8	$M _D N _E = (MN) _{D,E}$	
L.9	$M _{\lambda x.N} = M$	abstractions are total
L.10	$M _x = M$	variables are total
L.11	$(\lambda x.M)N = (M[x := N]) _N$	restricted β – reduction
L.12	$M = N \implies \lambda x.M = \lambda x.N$	ξ – rule

Table 1: Axioms and rules of λ^R

where $D[x := N]$ is shorthand for $\{P[x := N] \mid P \in D\}$.

The formulas of the theory λ^R are of the form $M = N$, where M, N are in Λ^R . The axioms and rules are displayed in Table 1.

The first five of these axioms are the same as for the ordinary lambda calculus and need no explanation. Axiom 6 says that restricting a term to itself does not change the term, since all the information about the domain of the term is already present. Axioms 7 and 8 say that repeated restriction can be flattened, that is, we can collect all the restrictions in one restriction bag. Restrictions behave well with respect to application, according to axiom 8. Axiom 9 states that lambda abstractions are total (in the remaining variables). Also, variables are total, but axiom 10 has to be read with the understanding that this doesn't mean we can simply forget about restrictions to variables: the equation stated is still an equation within the context of the variable x . See the substitution properties for more on this. The β -conversion axiom is the most important one: if we substitute N for x , we have to keep track of the definedness of N by putting N in our restriction bag. Finally, just as in the ordinary lambda calculus, we have the ξ -rule.

Remark. One could also define the partial lambda calculus *with constants*, by adding a constant c_a to the set of terms for each element a of some fixed set A . We will not need this construction, even though in many treatments of the subject, it is an essential ingredient. See the notes after section 5 for an explanation.

Lemma 3.2. *The theory λ^R has the following properties:*

1. $M = N \implies M|_D = N|_D$
2. $U = V \implies M|_U = M|_V$
3. If $x = x|_D$ then $M|_D = N|_D$ implies $M = N$

$$4. (\lambda y.M)[x := N] \equiv \lambda y.M[x := N]$$

5. *Provable equality respects substitution, in the sense that*

$$\lambda^R \vdash M = N \implies \lambda^R \vdash M[x := Q]_{|Q} = N[x := Q]_{|Q}.$$

6. *If M contains an occurrence of x which is not within the scope of a lambda abstraction, then $M[x := Q]_{|Q} = M[x := Q]$*

Proof. We use the rules and calculate:

1. If $M = N$, then by rule ξ , we have $\lambda x.M = \lambda x.N$ where x does not occur in M, N . Now $M_{|D_1} = (\lambda x.M)D_1 = (\lambda x.N)D_1 = N_{|D_1}$. By induction on the number of elements $D_i \in D$ the result follows.
2. If $U = V$, then again $M_{|U} = (\lambda x.M)U = (\lambda x.M)V = M_{|V}$.
3. $M = (\lambda x.x)M = (\lambda x.x_{|D})M = M_{|D} = N_{|D} = (\lambda x.x_{|D})N = (\lambda x.x)N = N$.
4. Induction on the structure of M .
5. Induction on the proof of $M = N$. When we look at the case $M_{|x} = M$, we observe that substitution gives $M[x := N]_{|N} = M[x := N]$, which is not a valid equation if x is not free in M and N is not total. The restriction to N is precisely to overcome this problem.
6. Induction on the structure of M .

□

The next lemma will not be used in the sequel, but may still be of some interest. It says that any term can be flattened, i.e. is provably equal to a term without restrictions.

Lemma 3.3. *For any term M , there is a term M' with $M = M'$ such that M' does not contain the symbol $|$ in any of its subterms*

Proof. Induction on the structure of M , the crucial case being $M = N_{|D}$, which is equal to $(\lambda u.N)D$, where u is fresh. □

There is also a “dual” result, saying that every term M can be saturated by adding each subterm of M to the restriction. These facts are useful when one wants to study the proof theory of the partial lambda calculus, see [Mog88].

To close this section, we mention that the theory λ^R is consistent. A cheap and uninformative way of seeing this is by adding the axiom $M_{|N} = M$ for all terms M, N , which amounts to saying that restrictions are irrelevant, or to saying that every term is total. Then we can translate λ^R into λ by simply forgetting restrictions. For the fact that the theory does not already prove $M_{|N} = M$ for all M, N , the reader is referred to the models we exhibit in later sections. In these models, the restriction will be non-trivial, showing that there are models of the partial lambda calculus which are not at the same time models of the total lambda calculus, and thus that the partial lambda calculus is genuinely different from the total lambda calculus.

3.2 Partial combinatory logic

We now describe a partial variant of combinatory logic. In essence, this is a special instance of a term logic for cartesian restriction categories. The theory will be equational, but terms can be partial, and the equality symbol will denote Kleene equality¹. Also, terms can be restricted, as in the partial lambda calculus. Just as in that theory, it is important to keep in mind that terms really are terms in a variable context, as are equations. In the notation, however, we do not keep track of all this information, since we work in a single-sorted system.

Definition 3.4 (Partial CL terms). *Let $Var = \{x_0, x_1, \dots\}$ be a countably infinite set of variables. The collection of terms of partial combinatory logic is denoted by CL^R and is inductively defined by:*

- $Var \subseteq CL^R$
- $\mathbf{k}, \mathbf{s} \in CL^R$
- $M, N \in CL^R \implies (MN) \in CL^R$
- $M \in CL^R, D \in CL^R \implies M|_D \in CL^R$

We use the same conventions for restrictions as in the partial lambda calculus, so that the restriction of a term can be any finite set of terms.

The notion of substitution is extended by defining

$$M|_D[x := N] = M[x := N]|_{D[x:=N]},$$

just as in the partial lambda calculus.

The theory CL^R has equality judgements of the form $M = N$, where M, N are CL^R -terms. The axioms and rules are listed in Table 2.

Note that in the axiom for \mathbf{k} , we don't just throw Q away, but put it in the restriction bag, preserving the information about its domain. The last axiom says that \mathbf{s} is total in the first two variables.

An easy consequence of the axioms is the following rule:

$$M = N, A = B \implies M|_A = N|_B.$$

To see why this is true, observe that $M = N$ implies $\mathbf{k}M = \mathbf{k}N$, and thus $M|_A = \mathbf{k}MA = \mathbf{k}NB = N|_B$.

Lemma 3.5. *Let M be a CL^R -term. Then $M|_N = M$ for every subterm N of M . Moreover, if y occurs in M , then $M[y := N] = M[y := N]|_N$.*

Proof. By induction on the structure of M . The statement is clear when M is a constant, a variable or one of \mathbf{k}, \mathbf{s} . If $M = PQ$ then we need to check the $M|_P = M$ and $M|_Q = M$. But $M|_P = (PQ)|_P = P|_P Q = PQ = M = PQ|_Q = M|_Q$. If

¹Traditionally, Kleene equality of terms t, t' is denoted by $t \simeq t'$, and is interpreted as: "When one of t, t' is defined, so is the other, in which case they are equal". Here, we will not use the symbol \simeq , but write $=$ for equality of partial terms.

CL.1	$M = M$	
CL.2	$M = N \implies N = M$	
CL.3	$M = N, N = P \implies M = P$	
CL.4	$M = N \implies MZ = NZ$	
CL.5	$M = N \implies ZM = ZN$	
CL.6	$M _M = M$	
CL.7	$M _{(N _D)} = M _{D,N}$	
CL.8	$M _D N _E = (MN) _{D,E}$	
CL.9	$M _x = M$	variables are total
CL.10	$M _k = M _s = M$	k, s are total
CL.11	$kPQ = P _Q$	reduction for k
CL.12	$sPQR = PR(QR)$	reduction for s
CL.13	$M _{sPQ} = M _{P,Q}$	s total in two arguments

Table 2: Axioms and rules for CL^R

$M = N|_D$ then we have $M|_N = M$ and for every $X \in D$ we have $M|_X = M$, so by induction we are done.

For the second claim, simply observe that if y occurs in M , then N is a subterm of $M[y := N]$, and apply the first part of the lemma. \square

In the total version of CL, provable equality is invariant under substitution. In the partial situation, we have to be a bit careful, since substituting for variables which do not occur in a term would result in a loss of information about the partiality of the term. The appropriate formulation is:

Lemma 3.6. *The theory CL^R satisfies the following restricted substitution properties:*

1. $M = N \implies M[x := A]|_A = N[x := A]|_A$
2. $A = B \implies M[x := A] = M[x := B]$
3. $M = N, A = B \implies M[x := A]|_A = N[x := B]|_B$.

Proof. The first item is proved by induction on the proof of $M = N$; the second by induction on the structure of M , and the third by combining the first two. \square

Remark. Just as for the partial lambda calculus, consistency of the theory CL^R is easily seen by adding the axiom $M|_N = M$; this essentially gives back ordinary combinatory logic. Non-trivial non-total models will be discussed in section 6.

3.3 Lambda abstraction and combinatory completeness

When first confronted with combinatory logic (or a partial version thereof), the axioms look cryptic and unintuitive. The point of the combinators k and s , however, is, that they form a basis for the key property of CL, namely combinatory

completeness. An essential ingredient is the fact that in partial combinatory logic one can mimick lambda abstraction. Formally, we define, for each variable x and for each CL^R -term M , a new term $\lambda^*x.M$. This is the content of the following definition.

Definition 3.7. *Let M be a CL^R -term and x a variable. Define $\lambda^*x.M$ by induction on the structure of M :*

- $\lambda^*x.x = \mathbf{skk}$
- $\lambda^*x.P = \mathbf{k}P$ if P is either \mathbf{k}, \mathbf{s} or a variable different from x
- $\lambda^*x.(MN) = \mathbf{s}(\lambda^*x.M)(\lambda^*x.N)$
- $\lambda^*x.M|_D = \mathbf{s}(\mathbf{s}(\mathbf{k}\mathbf{k})(\lambda^*x.M))(\lambda^*x.D)$

In the last clause, we assume that the restriction consists of a single term; if it consists of several terms, just iterate the construction. Thus, $\lambda^*x.M|_{D,E} = \lambda^*x.(M|_D)|_E$.

The following proposition sums up the virtues of the term just defined.

Proposition 3.8. *For a term M and a variable x we have:*

1. $FV(\lambda^*x.M) = FV(M) - \{x\}$
2. $\lambda^*x.M$ is total
3. $(\lambda^*x.M)N = M[x := N]|_N$

Proof. The first two items are a straightforward structural induction. For the third item, we show how to handle the new clause involving restriction. Let us verify that $(\lambda^*x.M|_D)N = (M|_D[x := N])|_N$.

$$\begin{aligned}
(\lambda^*x.M|_D)N &= (\mathbf{s}(\mathbf{s}(\mathbf{k}\mathbf{k})(\lambda^*x.M))(\lambda^*x.D))N \\
&= \mathbf{s}(\mathbf{k}\mathbf{k})(\lambda^*x.M)N((\lambda^*x.D)N) \\
&= (\mathbf{k}\mathbf{k})N((\lambda^*x.M)N)((\lambda^*x.D)N) \\
&= \mathbf{k}|_N((\lambda^*x.M)N)((\lambda^*x.D)N) \\
&= \mathbf{k}|_N M[x := N]|_N D[x := N]|_N \\
&= \mathbf{k}M[x := N]D[x := N]|_N \\
&= (M[x := N]|_N)|_{D[x := N]|_N} \\
&= M[x := N]|_{D[x := N], N} \\
&= (M|_D[x := N])|_N.
\end{aligned}$$

□

Some remarks are in order. First of all, the reader who is familiar with ordinary combinatory logic may wonder at this point why we don't define the clause $\lambda^*x.P = \mathbf{k}P$ whenever x does not occur in P . The reason is, that the

second item of the Proposition would fail. Indeed, take elements a, b such that ab is not total. Then $k(ab)$ is not total either. Since in the partial lambda calculus all lambda abstractions are total, we wish to have a corresponding property for the lambda abstraction operation in CL, and thus need to define this more complicated translation. For the relation between both translations, see the Appendix.

Second, the operation λ^*x is not well-behaved with respect to substitution. (This is the price one has to pay for using the more complicated translation.) That is, we don't have $(\lambda^*x.M)[y := N] = \lambda^*x.M[y := N]$, even if y does occur in M .

In spite of its shortcomings, the lambda abstraction is still very powerful, as it allows us to prove combinatory completeness:

Proposition 3.9 (Combinatory completeness of partial CL). *For every term M of CL^R with $FV(M) \subseteq \{x_1, \dots, x_{n+1}\}$ there is a term M' such that M' is closed, total in the first n arguments, and*

$$M'N_1 \cdots N_{n+1} = M[x_i := N_i]$$

for all terms N_1, \dots, N_{n+1} .

Proof. Take $M' = \lambda^*x_1 \cdots \lambda^*x_{n+1}.M$. □

3.4 Relation between CL^R and the partial λ -calculus

Having the lambda abstraction operation at our disposal, we get back to the relation between partial CL and the partial lambda calculus. The translations between the two are set up in the following definition.

Definition 3.10. *For a CL^R -term M , define a partial lambda term M_λ by induction on the structure of M .*

- $(x)_\lambda = x$
- $(MN)_\lambda = M_\lambda N_\lambda$
- $(M|_D)_\lambda = (M_\lambda)|_{D_\lambda}$
- $(\mathbf{k})_\lambda = \mathbf{S} := \lambda xy.x$
- $(\mathbf{s})_\lambda = \mathbf{K} := \lambda xyz.xz(yz)$

For a partial lambda term P , define a CL^R -term P_{CL} by induction on the structure of P :

- $(x)_{CL} = x$
- $(MN)_{CL} = M_{CL}N_{CL}$
- $(M|_D)_{CL} = (M_{CL})|_{D_{CL}}$

- $(\lambda x.M)_{CL} = \lambda^* x.M_{CL}$

Lemma 3.11. *The interpretation of CL^R in Λ^R is sound, i.e.*

$$CL^R \vdash M = N \implies \lambda^R \vdash M_\lambda = N_\lambda.$$

Proof. Straightforward induction on the proof of $M = N$. For the totality of \mathbf{s} , use the fact that lambda abstractions are total. \square

Lemma 3.12. *For all CL^R -terms $\lambda^R \vdash P_{CL,\lambda} = P$.*

Proof. We show first that $(\lambda^* x.M)_\lambda = \lambda x.M_\lambda$. This is done by induction on the structure of M . The crucial case is $M = N|_D$. We calculate:

$$\begin{aligned} (\lambda^* x.N|_D)_\lambda &= (\mathbf{s}(\mathbf{s}(\mathbf{kk})(\lambda^* x.N))(\lambda^* x.D))_\lambda \\ &= \mathbf{S}(\mathbf{S}(\mathbf{KK})(\lambda x.N_\lambda))(\lambda x.D_\lambda) \\ &= \lambda z.\mathbf{S}(\mathbf{KK})(\lambda x.N_\lambda)z((\lambda x.D_\lambda)z) \\ &= \lambda z.\mathbf{K}(\lambda x.N_\lambda)z((\lambda x.D_\lambda)z) \\ &= \lambda z.\mathbf{K}(N_\lambda[x := z])(D_\lambda[x := z]) \\ &= \lambda z.N_\lambda[x := z]|_{D_\lambda[x := z]} \\ &= \lambda x.N_\lambda|_{D_\lambda}. \end{aligned}$$

\square

We can specify a finite set of closed equations A_β^R , such that when these are added to CL^R , the above translations constitute an equivalence of theories. More precisely,

Theorem 3.13. *The theories $CL^R + A_\beta^R$ and λ^R are equivalent, in the following sense:*

1. $\lambda^R \vdash M = N \iff M_{CL,\lambda} = N_{CL,\lambda}$
2. $CL^R + A_\beta^R \vdash N = M \iff N_\lambda = M_\lambda$
3. $\lambda^R \vdash M = N \iff CL^R + A_\beta^R \vdash M_{CL} = N_{CL}$
4. $CL^R + A_\beta^R \vdash P = Q \iff \lambda^R \vdash P_\lambda = Q_\lambda$.

Because of its technical nature, the proof of this theorem (and, indeed, the formulation of the axiom set A_β^R) is relegated to the appendix.

3.5 Notes and references

Moggi's Ph. D. thesis [Mog88] is the main reference for logics and calculi for partiality. Among the various partial versions of the lambda calculus, the lambda calculus with a restriction operation is presented (there called the λ_p -calculus). This is essentially the system we presented here (although Moggi generally assumes the η -rule to be present, which we do not want here).

Our version of partial combinatory logic is, as mentioned in the text, nothing but an instance of the term logic for cartesian restriction categories. In Moggi's thesis one finds the precise connections between logics which are based on restrictions of terms and other formalisms for handling partiality, such as Beeson's logic of partial terms LPT (see [Bee85]) or Scott's logic of partial elements LPE (see [Sco79]).

Just as there is a correspondence between cartesian closed categories and typed lambda calculi (see [Lam86]), there is a correspondence between typed partial lambda calculi and partial cartesian closed categories (for a recent discussion, see [Sch04]). This can be extended to cartesian closed restriction categories, making precise in which sense (typed) partial lambda calculi are logics of ccrs.

In [Eve92], the authors prove that the partial lambda calculus (in a formulation which is almost the same as the one presented here) can be finitely axiomatized over combinatory logic with partial elements. Since our version of partial CL is more similar to the partial lambda calculus, in the sense that both are based on restriction, the comparison of the theories is a bit more smooth and perspicuous, since the translations are more straightforward. The proof of the finite axiomatization remains quite a bit of technical work, although in our setup, we can directly adapt the classical proof (for an account of these matters, see [Bar84]).

4 Combinatory structures in a restriction category

In this section we formulate the notion of a partial combinatory algebra inside a cartesian restriction category. Just as a total combinatory algebra is precisely a model of combinatory logic, a PCA in a restriction category is precisely a model of CL^R . An analogue of the classical observation that every total combinatory algebra is a Turing object in a certain category of representable maps (and vice versa) is obtained. We also define a notion of a partial lambda algebra in a restriction category. For the case where the restriction category is the category Par of sets and partial maps, this gives a strengthening of the notion of a PCA, just as the notion of a lambda algebra strengthens that of a combinatory algebra.

From now on, we will work in restriction categories, which have a built-in notion of partiality, and partiality will be the norm. Therefore, we will drop the qualifier “partial” in all of our definitions, and assume that everything we speak of is partial by default. For example, when we say “applicative structure” (see the next definition) we mean “partial applicative structure”. In order to refer to the limiting case where there is no nontrivial partiality, we use the adjective “total”. Thus, for example, by a “total combinatory algebra” we will understand what is ordinarily known as a combinatory algebra (or, slightly uncomfortably, a total PCA).

4.1 Combinatory Algebras

In a cartesian restriction category an **applicative structure** $A = (A, \bullet)$ is an object A together with a map $A \otimes A \xrightarrow{\bullet} A$ called **application**. The applicative system is said to be **total** if application is a total map.

This map give a series of maps

$$\begin{array}{ccc} A & \xrightarrow{\bullet^0=1_A} & A \\ A \otimes A & \xrightarrow{\bullet^1=\bullet} & A \\ & \vdots & \\ A^{\otimes n+2} \otimes A & \xrightarrow{\bullet^{n+1}=(\bullet^n \otimes 1)\bullet} & A. \end{array}$$

Thus, application is understood to be associating to the left. Since the arity of the operation can be recovered from the rest of the notation, we sometimes simply write $A^{\otimes n+1} \xrightarrow{\bullet} A$ for \bullet^n .

Definition 4.1. *Let $A = (A, \bullet)$ be an applicative system in \mathcal{C} .*

1. A map $f : A^{\otimes n} \rightarrow A$ is **A -computable** (or simply computable when A is understood) in case there is a total element $a_f : \top \rightarrow A$ such that the

following diagram commutes:

$$\begin{array}{ccc}
 A^{\otimes n} & \xrightarrow{f} & A \\
 a_f \otimes 1 \downarrow & \nearrow \bullet & \\
 A \otimes A^{\otimes n} & &
 \end{array}$$

Moreover, a_f is required to be total in its first $n - 1$ arguments.

2. A map $f : A^{\otimes n} \rightarrow A^{\otimes m}$ is **A -computable** in case $f = \langle g_1, \dots, g_m \rangle$ and each $g_i : A^{\otimes n} \rightarrow A$ is A -computable.

In case $m = 0$, this is supposed to be understood as saying that the unique total map into the terminal object is computable. In case $n = 0$, the definition merely states that the computable elements $1 \rightarrow A$ are precisely the total ones.

Definition 4.2. A map $A^{\otimes n} \rightarrow A$ is called **algebraic** if it can be built from projections and elements of A using application and compositions. This is extended coordinatewise to maps $A^{\otimes n} \rightarrow A^{\otimes m}$. An applicative system is called **combinatory complete** if the algebraic maps are exactly the computable ones.

As usual, combinatory completeness follows from two of its instances. An applicative system in a restriction category is a **combinatory algebra** in case it has two total elements $\mathbf{k}, \mathbf{s} : \top \rightarrow A$ which satisfy:

1. $(\mathbf{k} \bullet x) \bullet y = x$
2. $(\mathbf{s} \bullet x) \bullet y \bullet z = (x \bullet z) \bullet (y \bullet z)$
3. $\overline{(\mathbf{s} \otimes 1 \otimes 1) \bullet^2} = 1$

In this definition, the x, y are variables, so that the first two conditions can be expressed by commutativity of the diagrams:

$$\begin{array}{ccc}
 A \otimes A & \xrightarrow{\pi_0} & A \\
 k \otimes 1 \downarrow & \nearrow \bullet & \\
 A \otimes A \otimes A & &
 \end{array}
 \qquad
 \begin{array}{ccccccc}
 A^{\otimes 3} & \xrightarrow{1 \otimes 1 \otimes \Delta} & A^{\otimes 4} & \xrightarrow{\sigma_{0,2,1,3}} & A^{\otimes 4} & \xrightarrow{\bullet \otimes \bullet} & A \otimes A \xrightarrow{\bullet} A \\
 s \otimes 1 \downarrow & & & & & \nearrow \bullet & \\
 A^{\otimes 2} & & & & & &
 \end{array}$$

The top composite first duplicates the third argument, then permutes the second and third argument and then applies. Informally, the composite acts as $(x, y, z) \mapsto (x, y, z, z) \mapsto (x, z, y, z) \mapsto (xz, yz) \mapsto (xz)(yz)$.

The following proposition is now easily obtained using the methods from section 3.3.

Proposition 4.3. An applicative system (A, \bullet) is combinatory complete if and only if there are elements \mathbf{k}, \mathbf{s} making (A, \bullet) into a combinatory algebra.

Thus, the only difference (in our terminology) between a combinatory complete applicative system and a combinatory algebra is that in the latter a specific choice for the combinators \mathbf{k} and \mathbf{s} has been made. Such a choice is, in general, far from unique. An applicative system for which there is only one possible way of choosing \mathbf{k} and \mathbf{s} is sometimes called a **categorical** combinatory algebra, mainly by people who are in a particularly model-theoretic mindset.

It is important to note that a combinatory algebra in the restriction category \mathbf{Par} of sets and partial functions is precisely a PCA in the traditional sense. Another useful observation, which follows from the fact that all definitions only make use of cartesian restriction-theoretic concepts, is the following lemma, stating that cartesian restriction functors preserve combinatory algebras.

Lemma 4.4. *If A is an applicative system in a cartesian restriction category \mathbf{C} and $F : \mathbf{C} \rightarrow \mathbf{D}$ is a cartesian restriction functor, then F induces an applicative structure on $F(A)$ in \mathbf{D} . If A is a combinatory algebra, then so is $F(A)$.*

Concretely, the application on $F(A)$ is the map $F(\bullet) : F(A) \otimes F(A) \rightarrow F(A)$, and the combinators are $F(k), F(s)$.

An important example arises from observing that the total elements functor is cartesian, so that every combinatory algebra A in \mathbf{C} gives rise to a combinatory algebra $\Gamma_t(A)$ in the category \mathbf{Par} , which in turn gives an ordinary PCA in \mathbf{Set} .

4.2 Turing Objects

By definition, every combinatory algebra lives in some restriction category. We now wish to understand in which sense there can be different restriction categories housing the same combinatory algebra.

So let \mathbf{C} be a cartesian restriction category and A an applicative system A in \mathbf{C} . We say that A is a **Turing Object** in \mathbf{C} if every map $A^{\otimes n} \rightarrow A^{\otimes m}$ is A -computable. If, in addition, every object in \mathbf{C} is a retract of A , then we call \mathbf{C} a **Turing Category**. The central property of Turing objects is the following.

Proposition 4.5. *Let \mathbf{C} be a category with Turing object A . Then A is combinatory complete.*

Proof. One constructs algebraic maps corresponding to \mathbf{k} and \mathbf{s} as in the definition of a combinatory algebra, and concludes from the assumption that they are computable that representing elements \mathbf{k}, \mathbf{s} with the desired properties exist. \square

Thus, every Turing object is a combinatory algebra. However, not every combinatory algebra is a Turing object, because of the fact that there may be too many maps from A to A . For an example, just look at combinatory algebras in \mathbf{Par} , where the requirement that each partial function $A \rightarrow A$ is computable fails for cardinality reasons.

The following definition shows that we can always view a combinatory algebra as a Turing object in a certain Turing category.

Definition 4.6. *Let A be an applicative system in \mathbf{C} . Then $\mathbf{Comp}(\mathbf{C}, A)$ is the graph having as objects all powers of A and as maps the computable maps.*

In general, this graph will not have any good properties; for example, identities will not be present and maps will not compose. The following proposition relates properties of $\text{Comp}(\mathbf{C}, A)$ to properties of A .

Proposition 4.7. *The structure $\text{Comp}(\mathbf{C}, A)$ is a Turing category with Turing object A if and only if A is combinatory complete.*

Proof. By definition of $\text{Comp}(\mathbf{C}, A)$, every map $A^{\otimes n} \rightarrow A^{\otimes m}$ is A -computable. Combinatory completeness ensures that such maps are closed under composition and that identities are present. To show that every object $A^{\otimes n}$ is a retract of A , use suitable pairing functions (for the case $n = 0$, use a constant function). In addition, we have to show that $\text{Comp}(\mathbf{C}, A)$ is a restriction category. So let $f : A \rightarrow A$ be a computable map. We can define its restriction $\bar{f} : A \rightarrow A$ to be the algebraic map given by $kx(fx)$. It is straightforward to extend this to multiple variables. \square

The category $\text{Comp}(\mathbf{C}, A)$ comes equipped with an inclusion functor $i = i_{\mathbf{C}, A} : \text{Comp}(\mathbf{C}, A) \hookrightarrow \mathbf{C}$. This functor is cartesian, and therefore sends the Turing object A of $\text{Comp}(\mathbf{C}, A)$ to a combinatory algebra iA in \mathbf{C} . It is easily verified that this combinatory algebra is equal to the original combinatory algebra A that we started with.

Together, this shows that every combinatory algebra is a Turing object in a suitable category of computable maps. In a way that we will now make precise, $\text{Comp}(\mathbf{C}, A)$ is the universal solution to the problem of finding a Turing category in which A is a Turing object.

First, we define a category $\text{CA} - \text{Cat}$, where the objects are pairs (\mathbf{C}, A) consisting of a cartesian restriction category \mathbf{C} and a combinatory algebra A in \mathbf{C} . A morphism $(\mathbf{C}, A) \rightarrow (\mathbf{D}, B)$ is a cartesian restriction functor $F : \mathbf{C} \rightarrow \mathbf{D}$ such that $FA = B$. Two objects in the same path component are thought of as housing the same combinatory algebra.

We also define a category $\mathbf{T} - \text{Cat}$ which is the full subcategory of $\text{CA} - \text{Cat}$ on the objects (\mathbf{C}, A) , where \mathbf{C} is a Turing category with Turing object A . There is an evident inclusion functor

$$I : \mathbf{T} - \text{Cat} \rightarrow \text{CA} - \text{Cat}.$$

It is easily verified that when (\mathbf{C}, A) is a category with a combinatory algebra (Turing category), $K(\mathbf{C}, A) = (K(\mathbf{C}), A)$ is again a category with a combinatory algebra (Turing category). We write $\text{CA} - \text{Cat}_s$ for the subcategory of $\text{CA} - \text{Cat}$ on the split categories, and $\mathbf{T} - \text{Cat}_s$ for the full subcategory of $\mathbf{T} - \text{Cat}$ on the split Turing categories.

We can view the operation $A \mapsto \text{Comp}(\mathbf{C}, A)$ as a functor in the variable \mathbf{C} . Indeed, if we have a cartesian restriction functor $F : \mathbf{C} \rightarrow \mathbf{D}$, then we get an induced map $\text{Comp}(F, A) : \text{Comp}(\mathbf{C}, A) \rightarrow \text{Comp}(\mathbf{D}, FA)$ by sending an object $A^{\otimes n}$ to $FA^{\otimes n}$, and a computable map $f : A^{\otimes n} \rightarrow A^{\otimes m}$ to the map Ff , which is then again computable: if a represents f , then Fa represents Ff . Note that the inclusion $\text{Comp}(\mathbf{C}, A) \rightarrow \mathbf{C}$ is in fact part of a natural transformation from the functor Comp to the identity.

Proposition 4.8. *The functor $K \circ \text{Comp}$ is right adjoint to the inclusion $I : \mathbb{T} - \text{Cat}_s \rightarrow \text{CA} - \text{Cat}_s$. Here, K denotes the splitting of idempotents.*

Proof. The adjointness is induced by the inclusion functor $K(\text{Comp}(\mathbb{D}, B)) \hookrightarrow (\mathbb{D}, B)$. If we have a morphism F from a Turing category (\mathbb{C}, A) to an arbitrary (\mathbb{D}, B) , then we factor it through this inclusion by defining a cartesian restriction functor $\hat{F} : (\mathbb{C}, A) \rightarrow \text{Comp}(K(\mathbb{D}), B)$, as follows. For an arbitrary object X of \mathbb{C} , we know that X is a retract of A . Thus there is an inclusion map $m : X \rightarrow A$ and a retraction $r : A \rightarrow X$, and the composite idempotent mr on A is computable. Therefore, the map $F(mr)$ is computable in \mathbb{D} , so that we can view it as a morphism in $\text{Comp}(\mathbb{D}, B)$. When we split idempotents in the latter category, $\hat{F}(X) := F(mr)$ becomes an object in the Turing category $K(\text{Comp}(\mathbb{D}, B))$. This defines \hat{F} on objects; extension to morphisms is easy. \square

4.3 Combinatory algebras as models

In the previous section we formulated partial combinatory logic CL^R , and mentioned that it was in fact a special instance of a term logic for cartesian restriction categories. Therefore, it makes sense to speak of *models* of CL^R in a restriction category. Not surprisingly, this results in:

Proposition 4.9. *A model of CL^R in a cartesian restriction category \mathbb{C} is the same as a combinatory algebra in \mathbb{C} .*

The proof of this result is an immediate consequence of the fact that in an interpretation, variables are interpreted as projections, constants as elements, and restrictions as restriction idempotents.

We also mention that one can form a syntactical cartesian restriction category $\mathcal{G}(CL^R)$ containing a generic combinatory algebra. By this, we mean that there is a correspondence between combinatory algebras in \mathbb{C} (models of CL^R) and cartesian restriction functors $\mathcal{G}(CL^R) \rightarrow \mathbb{C}$. In fact, $\mathcal{G}(CL^R)$ will be a Turing category.

4.4 Partial lambda algebras

Let (A, \bullet) be an applicative system in \mathbb{C} . We wish to define when A is a (partial) lambda algebra. Of course, the results from the previous section allow us to do this, simply by requiring that it should be a model of the theory $CL^R + A_\beta^R$. (Remember that the axioms A_β^R made the theories CL^R and the partial lambda calculus equivalent.) However, that requirement would not only be very awkward to verify in practice, but also be missing the point, since we wish to think of a partial lambda algebra as a model of the partial lambda calculus. Therefore, we make an alternative definition in two steps. First, we say what it means to have an interpretation of the partial lambda calculus in A . Then, A is said to be a partial lambda algebra when this interpretation is sound, i.e. when provably equal terms get identified under the interpretation.

So, what should an interpretation of the partial lambda calculus in A be? We take a slightly more categorical approach than most authors, who typically, define when an applicative system in \mathbf{Set} is a lambda algebra. See the notes for some of the differences.

First some basic notation: when Δ is a set of variables, say $\Delta = \{x_1, \dots, x_n\}$, we write A^Δ for $A^{\otimes n}$, the n -fold product of A with itself. For two sets of variables $\Delta \subseteq \Theta$, we write π_Δ^Θ for the canonical projection $A^\Theta \rightarrow A^\Delta$. In case $\Delta = \{x\}$, we simply write $\pi_x^\Theta : A^\Theta \rightarrow A$. Finally, if $\bullet : A \otimes A \rightarrow A$ is the application, then for any pair of maps $\alpha, \beta : X \rightarrow A$, we write $\alpha \bullet \beta$ for the composite $\bullet \circ \langle \alpha, \beta \rangle : X \rightarrow A$.

Consider an assignment $\llbracket - \rrbracket_\Delta$ (for each variable context Δ) from partial lambda terms to maps in \mathbf{C} , where $\llbracket M \rrbracket_\Delta : A^\Delta \rightarrow A$ for all M with $FV(M) \subseteq \Delta$. In words, a term M in variable context Δ is interpreted as a morphism with codomain A , and where the domain is a product of A , with a copy of A for each variable in the context. We define $\llbracket - \rrbracket$ to be an **interpretation** if the following conditions are satisfied:

1. $\llbracket x \rrbracket_\Delta = \pi_x^\Delta : A^\Delta \rightarrow A$ when $x \in \Delta$
2. $\llbracket PQ \rrbracket_\Delta = \llbracket P \rrbracket_\Delta \bullet \llbracket Q \rrbracket_\Delta$
3. $\llbracket \lambda x.P \rrbracket_\Delta \bullet \llbracket M \rrbracket_\Delta = \llbracket P \rrbracket_{\Delta, x} \circ \langle \pi_\Delta^{\Delta, x}, \llbracket M \rrbracket_\Delta \rangle$
4. $\llbracket M \rrbracket_\Delta = \llbracket M \rrbracket_\Theta \circ \pi_\Delta^\Theta$ for $\Delta \subset \Theta$
5. $\llbracket M|_D \rrbracket_\Delta = \llbracket M \rrbracket_\Delta \circ \overline{\llbracket D \rrbracket_\Delta}$

The first clause says that variables are interpreted as projections, as is common in categorical logic. Next, application in the lambda calculus should be interpreted using the partial application map \bullet on A , as stated in the second clause. We don't have a direct way of stating *how* the lambda abstraction of a term should be interpreted (because of the fact that we don't work in a cartesian closed setting we don't have transposes available), but instead we say how it should *behave*. Indeed, when applying (the interpretation of) a term $\lambda x.P$ to (the interpretation of) another term M , the result should be the same as interpreting P in the context augmented by x , and then substituting M for x , where substitution is handled by means of composition. The fourth clause tells us how dummy variables should be handled: if we know how to interpret M in context Δ , and $\Delta \subseteq \Theta$, then the interpretation of M in context Θ is obtained by first projecting onto the A^Δ and then doing the interpretation in context Δ . Finally, the fifth clause says that restrictions of terms are to be interpreted using restriction idempotents.

Now we can state the key definition.

Definition 4.10. *Let (A, \bullet) be an applicative structure with interpretation $\llbracket - \rrbracket$. Then $(A, \bullet, \llbracket - \rrbracket)$ is called a **(partial) lambda algebra** if $\lambda^R \vdash M = N$ implies $\llbracket M \rrbracket = \llbracket N \rrbracket$.*

The condition in the definition is also expressed by saying that the interpretation of the partial lambda calculus is sound. The following proposition tells us that lambda algebras are indeed a strengthening of combinatory algebras, just as in the total case.

Proposition 4.11. *If $(A, \bullet, \llbracket - \rrbracket)$ is a lambda algebra, then A is combinatory complete.*

Proof. As expected, put $k = \llbracket \lambda xy.x \rrbracket$ and $s = \llbracket \lambda xyz.xz(yz) \rrbracket$. Soundness guarantees that $kab = a$ and $sabc = ac(bc)$. The fact that the interpretation of a lambda abstraction is always total ensures that s is total in the first two variables. \square

We postpone giving examples of lambda algebras in the partial setting, since this will be easier once we have the main results from the next section at our disposal. However, it is worth noting here that there are some important examples which distinguish PCAs from partial lambda algebras in the category of sets, such as Kleene's PCA of natural numbers with partial recursive application. Also, the generic model in the syntactic category $\mathcal{G}(CL^R)$ does not allow for a sound interpretation of the partial lambda calculus. Note that in order to even state these observations, we have to view these PCAs as combinatory algebras in restriction categories.

An important fact is that cartesian restriction functors preserve partial lambda algebras.

Proposition 4.12. *Let $F : C \rightarrow D$ be a cartesian restriction functor and let A be a lambda algebra in C . Then $F(A)$ is a lambda algebra.*

Proof. We already saw that cartesian restriction functors preserve combinatory algebras, and that the combinatory algebra structure on $F(A)$ is given by applying F to the structure on A . That is, the \mathbf{k} -combinator is given by $F(\mathbf{k})$, \mathbf{s} is given by $F(\mathbf{s})$, and application by $F(\bullet)$.

From Theorem 3.13 we know that a lambda algebra is a combinatory algebra which satisfies a finite number of extra axioms. It is therefore sufficient to show that F preserves the validity of these axioms. All the axioms are closed, that is, are of the form $M = N$ where M, N are simply words in \mathbf{k} and \mathbf{s} . As a consequence, the axioms of A_β^R correspond to equalities of elements $1 \rightarrow A$ built from \mathbf{k}, \mathbf{s} and \bullet . Since \mathbf{k}, \mathbf{s} , and application are preserved by F , so is the validity of the axioms. \square

Of course, it is possible to give an ad hoc proof of this proposition, by directly constructing the interpretation of lambda terms in $F(A)$, but this would be a lot more work.

4.5 Notes and references

For a good exposition of PCAs, basic properties, examples and constructions, see [Bet88]. For a different viewpoint, see Longley's thesis [Lon94].

There are some subtleties concerning the definition of a PCA and its meaning in categories different from \mathbf{Set} . First of all, it depends on one's point of view whether one wants the combinators \mathbf{k} and \mathbf{s} as part of the structure, or whether one wants the existence of these as a property. Typically, when considering PCAs as models of the theory of combinatory logic, one takes the structural point of view, and when PCAs are used for realizability purposes, one takes the property of combinatory completeness as fundamental. In the presence of the axiom of choice, one can, of course, always make a choice of combinators, but such a choice is usually far from unique. One possible objection to our (hybrid) definition could therefore be, that it assumes a certain amount of choice, and that it therefore excludes PCAs in toposes where even weak forms of choice fail. We can but hope that the offended reader is sufficiently compensated by the advantages given by our definition, such as the fact that only cartesian logic is needed and that, as a consequence, the quite useful result on preservation of PCAs and lambda algebras by cartesian functors comes for free.

For the total case, Turing categories and their connection to total combinatory algebras can be found in [Lon90]. Our notion of Turing category is a restriction category-theoretic version of the notion presented there, and some of the results presented here are straightforward partial versions of the results in loc. cit.

In their paper [Pao86], Di Paolo and Heller define the notion of a recursion category, which is similar to the Turing categories presented here, but not quite the same. For one thing, in a recursive category all objects are assumed to be isomorphic, whereas in a Turing category it is only required that every object is a retract of a universal object.

Partial lambda algebras already occur in Moggi's thesis [Mog88], and have been studied by Christian Even and Ramon Pino Pérez in their paper [Eve92]. Again, up to the choice of formal system to work in, our notion is essentially the same as theirs. For an overview of the theory of total lambda algebras, see [Bar84].

Typically, one defines a lambda algebra as an applicative system A with a sound interpretation of the lambda calculus. Our approach follows this general strategy, but our notion of interpretation is slightly different. Traditionally, one would consider lambda terms *with constants from A* , and interpret such lambda terms. Of course, one then requires that such constants are interpreted by the elements one started out with. Second, one usually handles variables using *valuations*. We use the fact that every applicative system lives in a category, and that we can use the logic of that category to interpret variables. Therefore, there is no need to consider the elements separately, and we have obtained an easy way to define a lambda algebra in any cartesian restriction category.

One aspect of the theory of models of the lambda calculus that we have not mentioned yet and will only touch upon later is the distinction between lambda algebras and lambda models, and the axiom of weak extensionality. See the remarks in section 5.3 and the references following that section.

5 Reflexive objects in a restriction category

In this section we present the generalization of the Scott-Koymans Theorem, which says that (i) every reflexive object in a cartesian closed category gives rise to a lambda algebra structure on the set of points, that (ii) every lambda algebra gives rise to a cartesian closed category with a reflexive object and that (iii) the lambda algebra induced by this reflexive object is isomorphic to the lambda algebra one started with. The generalization to partial ccc's first appeared in [Eve95], although a different formulation of partial combinatory logic and the partial lambda calculus was used there. The presentation below is a slight improvement in two respects: first, we are not confined to partial map categories. Second, we give a more informative analysis by making better use of the notion of a combinatory object inside a restriction category.

We start by explaining how every reflexive object in a cartesian closed restriction category \mathcal{C} is a partial lambda algebra in \mathcal{C} , and, as a consequence, that the total points form a partial lambda algebra in Par .

Next, we start with a lambda algebra in any cartesian restriction category \mathcal{C} , and construct a ccrc with reflexive object and a cartesian functor to $K(\mathcal{C})$ such that the partial lambda algebra structure on the reflexive object is, up to isomorphism, mapped by this functor to the original lambda algebra.

Most of the developments in this section follow the exposition in [Bar84], and many of the proofs given here are straightforward adaptations of the ones given in there.

5.1 Reflexive objects

We consider an object U in a ccrc \mathcal{C} , such that $[U \rightarrow U]$ is a total retract of U . That is, we have a diagram

$$[U \rightarrow U] \begin{array}{c} \xleftarrow{F} \\ \xrightarrow{G} \end{array} U, \quad F \circ G = 1_{[U \rightarrow U]}$$

in which the map F is total. In this situation, we say that U is a **reflexive object** in \mathcal{C} .

In this situation, we can define a map $\bullet : U \otimes U \rightarrow U$ as the composite $\epsilon \circ (G \otimes Id)$, as in

$$U \otimes U \xrightarrow{G \otimes Id} [U \rightarrow U] \otimes U \xrightarrow{\epsilon} U.$$

Here, ϵ is the evaluation map. For maps $f, g : X \rightarrow U$, we will again write $f \bullet g$ for the composite $\bullet \circ \langle f, g \rangle$. The following property of the application map will be used later.

Lemma 5.1. *Let $f, f' : X \rightarrow U$ be arbitrary maps, and let e, e' be restriction idempotents on X . Then*

$$(f \circ e) \bullet (f' \circ e') = (f \bullet f') \circ e \circ e'.$$

Proof. Calculate: $\langle fe, f'e' \rangle = (f \otimes f') \circ (e \otimes e') \circ \delta = (f \otimes f') \circ \delta \circ e \circ e'$, where the maps δ denote diagonals. \square

5.2 Interpretation of partial lambda terms

Traditionally, one shows that when U is a reflexive object in \mathbf{C} , the set $\Gamma(U)$ of elements of U is a lambda algebra. One would expect that we are going to generalize this by saying that a reflexive object U in a ccr makes $\Gamma_t(U)$, the set of total elements of U into a partial lambda algebra. However, because we defined the notion of a partial lambda algebra inside a cartesian restriction category, we can now take a more direct line: we will show that such a U is a partial lambda algebra in the ambient category \mathbf{C} . The fact that $\Gamma_t(U)$ is then a lambda algebra now is a simple consequence of the fact that cartesian restriction functors preserve lambda algebras.

To start, we need an interpretation of partial lambda terms. For $FV(M) = \{x_1, \dots, x_n\} \subseteq \Delta$, we will define

$$\llbracket M \rrbracket_\Delta : U^\Delta \rightarrow U,$$

where U^Δ stands for $U^{\otimes m}$ if $\Delta = \{x_1, \dots, x_m\}$. This will be done by induction on the structure of the term M :

- $\llbracket x_i \rrbracket_\Delta = \pi_i^\Delta : U^\Delta \rightarrow U$
- $\llbracket MN \rrbracket_\Delta = \llbracket M \rrbracket_\Delta \bullet \llbracket N \rrbracket_\Delta$
- $\llbracket \lambda x.M \rrbracket_\Delta = U^\Delta \xrightarrow{\mathbf{tr}(\llbracket M \rrbracket_{\delta,x})} [U \rightarrow U] \xrightarrow{G} U$
- $\llbracket M|_D \rrbracket_\Delta = \llbracket M \rrbracket_\Delta \circ \overline{\llbracket D \rrbracket_\Delta}$

In the third clause, $\mathbf{tr}(\llbracket M \rrbracket_{\delta,x})$ denotes the exponential transpose of $\llbracket M \rrbracket_{\delta,x} : U^{\Delta,x} = U^\Delta \otimes U \rightarrow U$.

We note that the interpretation of a variable is always a total map, since the projections in a ccr are total. It is also easy to see that the interpretation of a lambda abstraction $\lambda x.M$ is always total, since it is the composite of G and a transpose.

In order to prove that this defines an interpretation, note that the only condition which is not obvious from the definition is dealt with in the following lemma.

Lemma 5.2. *Let $FV(M) \subseteq \Delta \subseteq \Theta$. Then $\llbracket M \rrbracket_\Theta = \llbracket M \rrbracket_\Delta \circ \pi_\Delta^\Theta$.*

Proof. By induction on the structure of M .

- If $M = x$ then $\llbracket M \rrbracket_\Theta = \llbracket x \rrbracket_\Theta = \pi_x^\Theta = \pi_x^\Delta \circ \pi_\Delta^\Theta = \llbracket x \rrbracket_\Delta \circ \pi_\Delta^\Theta$.

- If $M = M_1M_2$, then

$$\begin{aligned}
\llbracket M_1M_2 \rrbracket_{\Theta} &= \epsilon \circ (G \otimes 1) \circ \langle \llbracket M_1 \rrbracket_{\Theta}, \llbracket M_2 \rrbracket_{\Theta} \rangle \\
&= \epsilon \circ (G \otimes 1) \circ \langle \llbracket M_1 \rrbracket_{\Delta}, \llbracket M_2 \rrbracket_{\Delta} \rangle \circ \pi_{\Delta}^{\Theta} \quad \text{by IH} \\
&= \llbracket M_1M_2 \rrbracket_{\Delta} \circ \pi_{\Delta}^{\Theta}.
\end{aligned}$$

- If $M = \lambda x.N$, then by induction hypothesis we have $\llbracket N \rrbracket_{\Theta, x} = \llbracket N \rrbracket_{\Delta, x} \circ \pi_{\Delta, x}^{\Theta, x}$. Therefore $tr(\llbracket N \rrbracket_{\Theta, x}) = tr(\llbracket N \rrbracket_{\Delta, x}) \circ \pi_{\Delta}^{\Theta}$, so that

$$\llbracket \lambda x.N \rrbracket_{\Delta} = G \circ tr(\llbracket N \rrbracket_{\Theta, x}) = G \circ tr(\llbracket N \rrbracket_{\Delta, x}) \circ \pi_{\Delta}^{\Theta} = \llbracket \lambda x.N \rrbracket_{\Theta} \circ \pi_{\Delta}^{\Theta}.$$

- If $M = N|_D$, then (assuming without loss of generality that D consists of one term only) we have

$$\begin{aligned}
\llbracket M|_D \rrbracket_{\Theta} &= \llbracket M \rrbracket_{\Theta} \circ \overline{\llbracket D \rrbracket_{\Theta}} \\
&= \llbracket M \rrbracket_{\Delta} \circ \pi_{\Delta}^{\Theta} \circ \overline{\llbracket D \rrbracket_{\Delta} \circ \pi_{\Delta}^{\Theta}} \quad \text{by IH} \\
&= \llbracket M \rrbracket_{\Delta} \circ \overline{\llbracket D \rrbracket_{\Delta} \circ \pi_{\Delta}^{\Theta}} \\
&= \llbracket M|_D \rrbracket_{\Delta} \circ \pi_{\Delta}^{\Theta}.
\end{aligned}$$

□

Now that we know that $\llbracket - \rrbracket$ is indeed an interpretation of partial lambda terms, we wish to establish that this interpretation is sound. This requires some technicalities with regard to substitution.

Lemma 5.3. *Let $FV(M) \subseteq \Delta = (x_1, \dots, x_n)$, let N_1, \dots, N_n fit in x_1, \dots, x_n , and $FV(N_1, \dots, N_n) \subseteq \Theta$. Then*

$$\llbracket M[\vec{x} := \vec{N}] \rrbracket_{\Theta} \circ \overline{\llbracket N_1 \rrbracket_{\Theta}} \circ \dots \circ \overline{\llbracket N_n \rrbracket_{\Theta}} = \llbracket M \rrbracket_{\Delta} \circ \langle \llbracket N_1 \rrbracket_{\Theta}, \dots, \llbracket N_n \rrbracket_{\Theta} \rangle.$$

In particular,

$$\llbracket M[\vec{x} := \vec{N}] \rrbracket_{\Theta} \geq \llbracket M \rrbracket_{\Delta} \circ \langle \llbracket N_1 \rrbracket_{\Theta}, \dots, \llbracket N_n \rrbracket_{\Theta} \rangle.$$

Proof. Again induction on the structure of M .

- If $M = x_i \in \Delta$, then $\llbracket M \rrbracket_{\Delta} = \pi_i^{\Delta}$, and $M[\vec{x} := \vec{N}] = N_i$, so that $\llbracket M[\vec{x} := \vec{N}] \rrbracket_{\Theta} = \llbracket N_i \rrbracket_{\Theta} = \llbracket N_i \rrbracket_{FV(N_i)} \circ \pi_{FV(N_i)}^{\Theta}$, by the previous lemma. Now we have:

$$\begin{aligned}
\llbracket M[\vec{x} := \vec{N}] \rrbracket_{\Theta} \circ \overline{\llbracket N_1 \rrbracket_{\Theta}} \circ \overline{\llbracket N_n \rrbracket_{\Theta}} &= \llbracket N_i \rrbracket_{\Theta} \circ \overline{\llbracket N_1 \rrbracket_{\Theta}} \circ \dots \circ \overline{\llbracket N_n \rrbracket_{\Theta}} \\
&= \pi_i^{\Delta} \circ \langle \llbracket N_1 \rrbracket_{\Theta}, \dots, \llbracket N_n \rrbracket_{\Theta} \rangle \\
&= \llbracket x_i \rrbracket_{\Delta} \circ \langle \llbracket N_1 \rrbracket_{\Theta}, \dots, \llbracket N_n \rrbracket_{\Theta} \rangle.
\end{aligned}$$

The first equation is precisely the defining property of the pairing in a cartesian restriction category.

- If $M = M_1M_2$ then, using the notation $\langle \llbracket N_i \rrbracket_{\Theta} \rangle = \langle \llbracket N_1 \rrbracket_{\Theta}, \dots, \llbracket N_n \rrbracket_{\Theta} \rangle$ and $\overline{\llbracket N_i \rrbracket_{\Theta}} = \overline{\llbracket N_1 \rrbracket_{\Theta}} \circ \dots \circ \overline{\llbracket N_n \rrbracket_{\Theta}}$:

$$\begin{aligned}
& \llbracket M[\vec{x} := \vec{N}] \rrbracket_{\Theta} \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \\
&= \llbracket M_1M_2[\vec{x} := \vec{N}] \rrbracket_{\Theta} \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \\
&= \llbracket M_1[\vec{x} := \vec{N}]M_2[\vec{x} := \vec{N}] \rrbracket_{\Theta} \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \\
&= (\llbracket M_1[\vec{x} := \vec{N}] \rrbracket \bullet \llbracket M_2[\vec{x} := \vec{N}] \rrbracket) \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \\
&= \epsilon \circ \langle F \circ \llbracket M_1[\vec{x} := \vec{N}] \rrbracket, \llbracket M_2[\vec{x} := \vec{N}] \rrbracket \rangle \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \\
&= \epsilon \circ \langle F \circ \llbracket M_1[\vec{x} := \vec{N}] \rrbracket \circ \overline{\llbracket N_i \rrbracket_{\Theta}}, \llbracket M_2[\vec{x} := \vec{N}] \rrbracket \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \rangle \\
&= \epsilon \circ \langle F \circ \llbracket M_1 \rrbracket_{\Delta} \circ \langle \llbracket N_i \rrbracket_{\Theta} \rangle, \llbracket M_2 \rrbracket_{\Delta} \circ \langle \llbracket N_i \rrbracket_{\Theta} \rangle \rangle \quad \text{by IH} \\
&= \epsilon \circ \langle F \circ \llbracket M_1 \rrbracket_{\Delta}, \llbracket M_2 \rrbracket_{\Delta} \rangle \circ \langle \llbracket N_i \rrbracket_{\Theta} \rangle \\
&= (\llbracket M_1 \rrbracket_{\Delta} \bullet \llbracket M_2 \rrbracket_{\Delta}) \circ \langle \llbracket N_i \rrbracket_{\Theta} \rangle \\
&= \llbracket M_1M_2 \rrbracket_{\Delta} \circ \langle \llbracket N_i \rrbracket_{\Theta} \rangle.
\end{aligned}$$

- If $M = \lambda y.N$, then

$$\begin{aligned}
& \llbracket (\lambda y.P)[\vec{x} := \vec{N}] \rrbracket_{\Theta} \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \\
&= \llbracket (\lambda y.P)[\vec{x} := \vec{N}, y := y] \rrbracket_{\Theta} \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \\
&= G \circ tr \left(\llbracket P[\vec{x} := \vec{N}, y := y] \rrbracket_{\Theta, y} \right) \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \\
&= G \circ tr \left(\llbracket P[\vec{x} := \vec{N}, y := y] \rrbracket_{\Theta, y} \circ (\overline{\llbracket N_i \rrbracket_{\Theta}} \otimes 1) \right) \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \\
&= G \circ tr \left(\llbracket P[\vec{x} := \vec{N}, y := y] \rrbracket_{\Theta, y} \circ (\overline{\llbracket N_i \rrbracket_{\Theta}} \otimes \overline{\llbracket y \rrbracket_y}) \right) \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \\
&= G \circ tr \left(\llbracket P[\vec{x} := \vec{N}, y := y] \rrbracket_{\Theta, y} \circ (\overline{\llbracket N_i \rrbracket_{\Theta, y}} \circ \overline{\llbracket y \rrbracket_{\Theta, y}}) \right) \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \\
&= G \circ tr \left(\llbracket P \rrbracket_{\Delta, y} \circ \langle \llbracket N_i \rrbracket_{\Theta, y}, \llbracket y \rrbracket_{\Theta, y} \rangle \right) \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \quad \text{by IH} \\
&= G \circ tr \left(\llbracket P \rrbracket_{\Delta, y} \circ (\langle \llbracket N_i \rrbracket_{\Theta} \rangle \otimes 1) \right) \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \\
&= G \circ tr \left(\llbracket P \rrbracket_{\Delta, y} \right) \circ \langle \llbracket N_i \rrbracket_{\Theta} \rangle \circ \overline{\llbracket N_i \rrbracket_{\Theta}} \\
&= G \circ tr \left(\llbracket P \rrbracket_{\Delta, y} \right) \circ \langle \llbracket N_i \rrbracket_{\Theta} \rangle \\
&= \llbracket \lambda y.P \rrbracket_{\Delta} \circ \langle \llbracket N_i \rrbracket_{\Theta} \rangle.
\end{aligned}$$

Here we used that from the equations $\langle \llbracket N_i \rrbracket_{\Theta, y}, \llbracket y \rrbracket_{\Theta, y} \rangle = \langle \llbracket N_i \rrbracket_{\Theta} \rangle \otimes 1$ and $\overline{\llbracket y \rrbracket_y} = 1$ it follows that $\langle \overline{\llbracket N_i \rrbracket_{\Theta, y}}, \overline{\llbracket y \rrbracket_{\Theta, y}} \rangle = \overline{\llbracket N_i \rrbracket_{\Theta}} \otimes 1$.

- Finally, take $M = P|_D$. We can again assume that D consists of a single

term, since then we can do an induction on the number of elements of D .

$$\begin{aligned}
& \llbracket M[\vec{x} := \vec{N}]_{|D[\vec{x} := \vec{N}]} \rrbracket_{\Theta} \circ \overline{\llbracket \vec{N} \rrbracket_{\Theta}} \\
&= \llbracket M[\vec{x} := \vec{N}] \rrbracket_{\Theta} \circ \overline{\llbracket D[\vec{x} := \vec{N}] \rrbracket_{\Theta} \circ \llbracket \vec{N} \rrbracket_{\Theta}} \\
&= \llbracket M[\vec{x} := \vec{N}] \rrbracket_{\Theta} \circ \overline{\llbracket \vec{N} \rrbracket_{\Theta}} \circ \overline{\llbracket D[\vec{x} := \vec{N}] \rrbracket_{\Theta} \circ \llbracket \vec{N} \rrbracket_{\Theta}} \\
&= \llbracket M[\vec{x} := \vec{N}] \rrbracket_{\Theta} \circ \overline{\llbracket \vec{N} \rrbracket_{\Theta}} \circ \overline{\llbracket D[\vec{x} := \vec{N}] \rrbracket_{\Theta} \circ \llbracket \vec{N} \rrbracket_{\Theta}} \\
&= \llbracket M \rrbracket_{\Delta} \circ \langle \llbracket \vec{N} \rrbracket_{\Theta} \rangle \circ \overline{\llbracket D \rrbracket_{\Theta} \circ \langle \llbracket \vec{N} \rrbracket_{\Theta} \rangle} \quad \text{by IH} \\
&= \llbracket M \rrbracket_{\Delta} \circ \overline{\llbracket D \rrbracket_{\Delta}} \circ \langle \llbracket \vec{N} \rrbracket_{\Theta} \rangle \\
&= \llbracket M_{|D} \rrbracket_{\Delta} \circ \langle \llbracket \vec{N} \rrbracket_{\Theta} \rangle.
\end{aligned}$$

□

Lemma 5.4. *Let $FV(\lambda x.M) \subseteq \Delta$, $FV((\lambda x.M)N) \subseteq \Theta$ and $\Delta \subseteq \Theta$. Then*

$$\llbracket M[x := N] \rrbracket_{\Theta} \circ \overline{\llbracket N \rrbracket_{\Theta}} = \llbracket M \rrbracket_{\Delta, x} \circ \langle \pi_{\Delta}^{\Theta}, \llbracket N \rrbracket_{\Theta} \rangle.$$

In particular,

$$\llbracket M[x := N] \rrbracket_{\Theta} \geq \llbracket M \rrbracket_{\Delta, x} \circ \langle \pi_{\Delta}^{\Theta}, \llbracket N \rrbracket_{\Theta} \rangle.$$

Proof. Put $\Theta = (\Delta, x)$ and $\vec{y} = \Delta$. Then $M[x := N] = M[x := N, \vec{y} := \vec{y}]$. Apply the previous lemma to find

$$\begin{aligned}
\llbracket M[x := N, \vec{y} := \vec{y}] \rrbracket_{\Theta} \circ \overline{\llbracket N \rrbracket_{\Theta}} &= \llbracket M[x := N, \vec{y} := \vec{y}] \rrbracket_{\Theta} \circ \overline{\llbracket N \rrbracket_{\Theta}} \circ \overline{\llbracket \vec{y} \rrbracket_{\Theta}} \\
&= \llbracket M \rrbracket_{\Theta} \circ \langle \llbracket \vec{y} \rrbracket_{\Theta}, \llbracket N \rrbracket_{\Theta} \rangle \\
&= \llbracket M \rrbracket_{\Theta} \circ \langle \pi_{y_1}^{\Theta}, \dots, \pi_{y_n}^{\Theta}, \llbracket N \rrbracket_{\Theta} \rangle \\
&= \llbracket M \rrbracket_{\Theta} \circ \langle \pi_{\Delta}^{\Theta}, \llbracket N \rrbracket_{\Theta} \rangle.
\end{aligned}$$

□

We can now prove that the interpretation $\llbracket - \rrbracket$ is sound for the partial lambda calculus.

Proposition 5.5 (Soundness). *The interpretation $\llbracket - \rrbracket$ is sound for the partial lambda calculus, meaning that $\lambda^R \vdash M = N$ implies $\llbracket M \rrbracket_{\Delta} = \llbracket N \rrbracket_{\Delta}$ whenever $FV(MN) \subseteq \Delta$.*

Proof. We do this by induction on the proof of $\lambda^R \vdash M = N$. Soundness is obvious for the first three axioms, which state that $=$ is an equivalence relation. Axioms 4 and 5 go through because of the way application is defined. Axioms 6, 7 and 8 translate into obvious truisms about idempotents. Soundness of axiom 9 follows from Lemma 5.1; The interpretation of a lambda abstraction is

a transpose and therefore total. As for β -reduction, we calculate

$$\begin{aligned}
\llbracket (\lambda x.P)Q \rrbracket_{\Delta} &= (G \circ \text{tr}(\llbracket P \rrbracket_{\Delta,x})) \circ \llbracket Q \rrbracket_{\Delta} \\
&= \epsilon \circ \langle F \circ G \circ \text{tr}(\llbracket P \rrbracket_{\Delta,x}), \llbracket Q \rrbracket_{\Delta} \rangle \\
&= \epsilon \circ \langle \text{tr}(\llbracket P \rrbracket_{\Delta,x}), \llbracket Q \rrbracket_{\Delta} \rangle \\
&= \epsilon \circ (\text{tr}(\llbracket P \rrbracket_{\Delta,x}) \otimes 1) \circ \langle 1, \llbracket Q \rrbracket_{\Delta} \rangle \\
&= \llbracket P \rrbracket_{\Delta,x} \circ \langle 1, \llbracket Q \rrbracket_{\Delta} \rangle \\
&= \llbracket P[x := Q] \rrbracket_{\Delta} \circ \overline{\llbracket Q \rrbracket_{\Delta}} \quad \text{by the previous lemma.}
\end{aligned}$$

Finally, the interpretation is sound for the ξ -rule because

$$\begin{aligned}
\llbracket P \rrbracket_{\Delta} = \llbracket Q \rrbracket_{\Delta} &\Rightarrow \llbracket P \rrbracket_{\Delta} \circ \pi_{\Delta}^{\Delta,x} = \llbracket Q \rrbracket_{\Delta} \circ \pi_{\Delta}^{\Delta,x} \\
&\Rightarrow \llbracket P \rrbracket_{\Delta,x} = \llbracket Q \rrbracket_{\Delta,x} \\
&\Rightarrow \llbracket \lambda x.P \rrbracket_{\Delta} = \llbracket \lambda x.Q \rrbracket_{\Delta}.
\end{aligned}$$

□

Our soundness theorem gives at once:

Theorem 5.6. *Every reflexive object U in a cartesian closed restriction category is a lambda algebra.*

As a corollary, we obtain:

Corollary 5.7. *The set of total points $\Gamma_t(U)$ is a lambda algebra in the category Par of sets and partial functions.*

Proof. The object U is a lambda algebra, and Γ_t is a cartesian restriction functor, so by Proposition 4.12, the object $\Gamma_t(U)$ is a lambda algebra. □

5.3 From partial lambda algebras to reflexive objects

We now embark on the generalization of the construction of a reflexive object from a lambda algebra. Given a PCA A for which the interpretation of lambda calculus is sound in the restricted sense described above, we can define, for any two elements $a, b \in A$:

$$a \diamond b = \llbracket \lambda x.a(bx) \rrbracket.$$

This should not be confused with either the notation \circ , which denotes arrow composition, or \bullet , which denotes partial application. When we write a juxtaposition, this is shorthand for application. Also, we will stop writing Scott-brackets everywhere in order to lighten up on the notation.

Note that $a \diamond b$ is always total, since interpretations of lambda abstraction terms are total. The following is now standard.

Lemma 5.8. *The set $M(A) = \{a \in A \mid \llbracket \lambda x.ax \rrbracket = a\}$ is a monoid under \diamond , with unit I .*

Proof. Use the soundness lemma and the rules for the partial lambda calculus. \square

But $M(A)$ is more than just a monoid. When A is not total, $M(A)$ has a nontrivial restriction structure, given by

$$\bar{a} = \lambda x.x|_{ax}.$$

Clearly, $\bar{a}x \downarrow$ if and only if $ax \downarrow$, in which case $\bar{a}x = x$.

Lemma 5.9. *The assignment $a \mapsto \bar{a}$ defines a restriction on $M(A)$.*

Proof. This is a straightforward application of the soundness theorem. Observe that $\bar{a}x = x|_{ax}$, and that $\overline{\bar{b}x} = \bar{b}x$.

For axiom (R.1), we calculate:

$$a \diamond \bar{a} = \lambda x.a(\bar{a}x) = \lambda x.a(x|_{ax}) = \lambda x.ax|_{ax} = \lambda x.ax = a.$$

For axiom (R.2), we have:

$$\bar{a} \diamond \bar{b} = \lambda x.\bar{a}(\bar{b}x) = \lambda x.(\lambda y.y|_{ay})(\bar{b}x) = \lambda x.\bar{b}x|_{a\bar{b}x} = \lambda x.x|_{ax,bx},$$

which, by symmetry, must equal $\bar{b} \diamond \bar{a}$.

To check the validity of axiom (R.3), consider:

$$\begin{aligned} \bar{a} \diamond \bar{b} &= \lambda x.\bar{a}(\bar{b}x) \\ &= \lambda x.x|_{ax|bx} \\ &= \lambda x.x|_{(\lambda z.az|bz)x} \\ &= \overline{\lambda x.ax|bx} \\ &= \overline{\lambda x.a(x|bx)} \\ &= \overline{\lambda x.a(\bar{b}x)} \\ &= \overline{a \diamond \bar{b}}. \end{aligned}$$

Finally we check axiom (R.4):

$$\begin{aligned} \bar{a} \diamond b &= \lambda x.\bar{a}(bx) \\ &= \lambda x.(\lambda z.z|_{az})(bx) \\ &= \lambda x.bx|_{a(bx)} \\ &= \lambda x.b(x|_{a(bx)}) \\ &= \lambda x.b((\lambda z.z|_{a(bz)})x) \\ &= \lambda x.b(\overline{(a \diamond \bar{b})}x) \\ &= b \diamond \overline{a \diamond \bar{b}}. \end{aligned}$$

\square

Now define $\mathbf{C}(A)$ to be the restriction category obtained by splitting all idempotents in $M(A)$. In this category, an object is an element $a = 1a$ such that $a \diamond a = a$, i.e. $\lambda x.a(ax) = \lambda x.ax = a$. A morphism $f : a \rightarrow b$ is an element $f = 1f$ such that $b \diamond f \diamond a = f$, i.e. $\lambda x.a(f(bx)) = f$. For such a morphism $f : a \rightarrow b$, the restriction is defined to be $\bar{f} \diamond a$, i.e. $\lambda x.ax|_{f(ax)}$.

We will make use of the following auxiliary notions. First, there is a combinator **Pair** in A , which is defined as $\mathbf{Pair} = \lambda xyz.zyx$. For terms or elements M, N , we write

$$[M, N] = \mathbf{Pair}MN = (\lambda z.zMN)|_{M,N}.$$

Note that this is the usual pairing, but restricted to the domains of the terms M and N . There are unpairings, defined by

$$\pi_0 = \lambda x.x\mathbf{True}, \quad \pi_1 = \lambda x.x\mathbf{False}.$$

Here, $\mathbf{True} := \lambda xy.x$ and $\mathbf{False} := \lambda xy.y$. These have the property that $\pi_0[M, N] = M|_N$ and $\pi_1[M, N] = N|_M$. We will often abbreviate $\pi_i(x) = x_i$. Also, we note that $[M|_N, N|_M] = [M, N]$.

Lemma 5.10. *The category $\mathbf{C}(A)$ is a cartesian restriction category.*

Proof. We start by exhibiting the terminal object. This is the idempotent $\mathbf{False} = \lambda xy.y$. If f is a map $a \rightarrow \mathbf{False}$ then $f = \mathbf{False} \diamond f \diamond a = \lambda u.\mathbf{False}(f(au)) = \lambda uy.y|_{f(au)} = \lambda uy.y|_{fu}$, and therefore f must be a restriction of \mathbf{False} . Hence, \perp is the unique total map $a \rightarrow \mathbf{False}$.

Next, we show that binary partial products exist. Given two idempotents a, b , define

$$a \otimes b = \lambda z.[az_0, bz_1], \quad p_0 = a \diamond \pi_0 \diamond (a \otimes b), \quad p_1 = b \diamond \pi_1 \diamond (a \otimes b).$$

To verify that $(a \otimes b)$ is indeed an idempotent, note that

$$\begin{aligned} (a \otimes b) \diamond (a \otimes b) &= \lambda x.(\lambda z.[az_0, bz_1])[ax_0, bx_1] \\ &= \lambda x.[a(ax_0)|_{bx_1}, b(bx_1)|_{ax_0}] \\ &= \lambda x.[ax_0|_{bx_1}, bx_1|_{ax_0}] \\ &= \lambda x.[ax_0, bx_1] \\ &= a \otimes b. \end{aligned}$$

It is obvious that p_0 and p_1 , as defined, are maps between idempotents. We verify that they are total:

$$\begin{aligned} p_0 &= a \diamond \pi_0 \diamond (a \otimes b) \\ &= \lambda x.a(\pi_0[ax_0, bx_1]) \\ &= \lambda x.a(ax_0)|_{bx_1} \\ &= \lambda x.ax_0|_{bx_1}, \end{aligned}$$

so that

$$\begin{aligned}
\overline{p_0} \diamond (a \otimes b) &= \lambda x. \overline{p_0}[ax_0, bx_0] \\
&= \lambda x. (\lambda y. y|_{ay_0, by_1})[ax_0, bx_1] \\
&= \lambda x. [ax_0, bx_1]|_{ax_0|bx_1, bx_1|ax_0} \\
&= \lambda x. [ax_0, bx_1]|_{ax_0, bx_1} \\
&= a \otimes b
\end{aligned}$$

and similarly for p_1 .

Then, given maps $f : c \rightarrow a$ and $g : c \rightarrow b$, define

$$\langle f, g \rangle = \lambda z. [fz, gz].$$

We have

$$\begin{aligned}
\langle f, g \rangle \diamond c &= \lambda x. \langle f, g \rangle(cx) \\
&= \lambda x. [f(cx), g(cx)]|_{cx} \\
&= \lambda x. [f(cx), g(cx)] \\
&= \lambda x. [fx, gx]
\end{aligned}$$

where we have used $f(cx)|_{cx} = f(cx)$ to get rid of the restriction. Also, we have

$$\begin{aligned}
(a \otimes b) \diamond \langle f, g \rangle &= \lambda x. (\lambda z. [az_0, bz_1])[fx, gx] \\
&= \lambda x. [a(fx)|_{gx}, b(gx)|_{fx}] \\
&= \lambda x. [a(fx), b(gx)] \\
&= \lambda x. [fx, gx] \\
&= \langle f, g \rangle.
\end{aligned}$$

Thus, the map $\langle f, g \rangle$ is indeed a map $c \rightarrow a \otimes b$.

To see that the map $\langle f, g \rangle$ has the correct domain, we calculate

$$\begin{aligned}
\overline{\langle f, g \rangle} \diamond c &= \lambda u. cu|_{[f(cu), g(cu)]} \\
&= \lambda u. cu|_{f(cu), g(cu)}.
\end{aligned}$$

Because $\overline{f} \diamond c = \lambda u. cu|_{f(cu)}$ and $\overline{g} \diamond c = \lambda u. cu|_{g(cu)}$, we see that the domain of $\langle f, g \rangle$ is indeed the intersection of the domains of f and g .

Finally, we have

$$\begin{aligned}
p_0 \diamond \langle f, g \rangle &= a \diamond \pi_0 \diamond (a \otimes b) \diamond \langle f, g \rangle \\
&= a \diamond \pi_0 \diamond \langle f, g \rangle \\
&= \lambda x. a(\pi_0[fx, gx]) \\
&= \lambda x. a(fx|_{gx}) \\
&= \lambda x. a(fx)|_{gx} \\
&= \lambda x. fx|_{gx}.
\end{aligned}$$

□

Lemma 5.11. *The category $\mathbf{C}(A)$ is a cartesian closed restriction category.*

Proof. Define the exponent of a and b as

$$[a \multimap b] = \lambda x. a \diamond x \diamond b$$

and the evaluation map $\epsilon_{a,b} : [a \multimap b] \otimes a \rightarrow b$ as

$$\epsilon_{a,b} = \lambda x. b(x_0(ax_1)).$$

If $f : c \otimes a \rightarrow b$, then define the transpose of f to be

$$\text{tr}(f) = \lambda xy. f[x, y] : c \rightarrow [a \multimap b].$$

Note that $\text{tr}(f)$ is total in x . The verifications that these definitions are correct don't contain any difficulties and are left to the reader. \square

Now we can state the anticipated result.

Theorem 5.12. *The object $I = \lambda x. x$ is a reflexive object in $\mathbf{C}(A)$.*

Proof. We have $[I \multimap I] = \lambda x. I \diamond x \diamond I = \lambda xy. I(x(Iy)) = \lambda xy. xy = 1$, which is an idempotent (on the unique object) and therefore a retract (of the unique object) in the category $\mathbf{C}(A)$. Moreover, the retraction map $1 : I \rightarrow 1$ is total, as required. \square

How does the reflexive object in $\mathbf{C}(A)$ relate to the original lambda algebra A in \mathbf{C} ? The following diagram clarifies the situation:

$$\begin{array}{ccc} M(A) & \longrightarrow & K(M(A)) = \mathbf{C}(A) \\ \downarrow Q & & \downarrow K(Q) \\ \text{Comp}(\mathbf{C}, A) & \longrightarrow & K(\text{Comp}(\mathbf{C}, A)) \\ \downarrow & & \downarrow \\ \mathbf{C} & \longrightarrow & K(\mathbf{C}) \end{array}$$

The three horizontal maps are the inclusions of $M(A)$, $\text{Comp}(\mathbf{C}, A)$ and \mathbf{C} in the corresponding split categories. Remember that $\text{Comp}(\mathbf{C}, A)$ is the (non-full) subcategory of \mathbf{C} on the powers of A and the computable maps, and that, by construction, this inclusion is a cartesian functor. Therefore it induces a cartesian functor on the level of split categories, making the bottom square commutative.

The functor Q is defined by sending the unique object of $M(A)$ to A , and by sending an element a of the monoid to the map represented by a , which we will denote by $Q(a) = a \bullet -$. This is functorial, since the map $(a \diamond b) \bullet - = \lambda x. a(bx) \bullet - = a(b-)$ is indeed the composite of $a \bullet -$ and $b \bullet -$. However, the monoid $M(A)$ is generally not cartesian as a restriction category (essentially because pairing need not be surjective), and therefore it makes no sense to ask for more properties of Q . But since $K(M(A))$ is cartesian, we can look at the induced restriction functor $K(Q)$ and prove:

Theorem 5.13. *The functor $K(Q)$ is cartesian, and induces an isomorphism of lambda algebras $K(Q)(A, I) \cong A$ in the category $K(\text{Comp}(\mathbf{C}, A))$, and hence also in the category $K(\mathbf{C})$.*

Proof. We first show that the functor $K(Q)$ preserves products. Given two idempotents a, b , we have the objects $(A, a \bullet -)$ and $(A, b \bullet -)$, and we have to show that their product is isomorphic to $(A, (a \otimes b) \bullet -)$.

It is clear that the two projections $p_0 : a \otimes b \rightarrow a$ and $p_1 : a \otimes b \rightarrow b$ represent a map $(A, (a \otimes b) \bullet -) \rightarrow (A, a \bullet -) \otimes (A, b \bullet -)$. We will show that the map $\gamma = \gamma_{a,b} = \lambda xy.[ax, by]$ represents an inverse to this map. Note first that $(a \otimes b)(\gamma xy) = (\lambda z.[az_0, bz_1])[ax, by] = [aax|_{by}, bby|_{ax}] = [ax, by] = \gamma xy$, and that $p_0 x = (a \diamond \pi_0 \diamond a \otimes b)x = a(\pi_0[ax_0, bx_1]) = ax_0|_{bx_1}$ and $p_1 x = bx_1|_{ax_0}$. Now

$$\gamma \bullet (p_0 x) \bullet (p_1 x) = [p_0 x, p_1 x] = [ax_0, bx_1] = (a \otimes b)x.$$

Also,

$$p_0(\gamma xy) = a(\pi_0(a \otimes b[ax, by])) = a(\pi_0[ax, by]) = aax|_{by},$$

so that $\langle \gamma(p_0 \bullet -), \gamma(p_1 \bullet -) \rangle = \langle ax \bullet -, by \bullet - \rangle$. This shows that γ indeed represents a two-sided inverse to $\langle p_0 \bullet -, p_1 \bullet - \rangle$. This isomorphism is readily extended to multiple factors.

There is also an isomorphism $(A, \perp) \cong \top$, showing that the terminal object is preserved. To see this, observe that a map $u : \top \rightarrow (A, \perp)$ must satisfy $u(*) = (\lambda xy.y)u(*) = I$, whence u is unique and gives an inverse to the map $(A, \perp) \rightarrow \top$.

Next, in the category $K(M(A))$ we have an application Ap on I , given by

$$I \otimes I \xrightarrow{1 \otimes I} [I \rightarrow I] \otimes I \xrightarrow{\epsilon} I.$$

Unwinding the definitions gives

$$Ap = \epsilon \diamond (1 \otimes I) = \lambda x.\epsilon((1 \otimes I)(x)) = \lambda x.\epsilon[1x_0, x_1] = \lambda x.(1x_0)(x_1) = \lambda x.x_0x_1.$$

Thus we have a commutative diagram in $\text{Comp}(\mathbf{C}, A)$:

$$\begin{array}{ccc} (A, I) \otimes (A, I) & \xrightarrow{\bullet} & (A, I) \\ \gamma \downarrow \cong & & \uparrow \epsilon \bullet - \\ (A, I \otimes I) & \xrightarrow{(1 \otimes I) \bullet -} & (A, 1 \otimes I) \end{array}$$

which shows that $K(Q)(I)$ is the same applicative structure as (A, I) . Finally, the interpretation of lambda terms is essentially the same, in the sense that for each term $M \in \Lambda_R(A)$, we have a commutative diagram

$$\begin{array}{ccc} (A, I) \otimes \cdots \otimes (A, I) & \xrightarrow{[M]_\Delta} & (A, I) \\ \gamma \downarrow & \nearrow K(Q)([M]_\Delta) & \\ (A, I \otimes \cdots \otimes I) & & \end{array}$$

Here, we have written $\llbracket M \rrbracket$ twice, once for the interpretation of M in the original lambda algebra A , and then for the interpretation of M in the reflexive object I in the category $K(M(A))$. The proof of this fact is by induction on the structure of M .

Alternatively, one can invoke Proposition 4.12. □

It is important to note the role that the composite $I \xrightarrow{1} [I \rightarrow I] \xrightarrow{1} I$ plays: Since $1 \diamond 1 = 1$, we derive that $I \cong [I \rightarrow I]$ precisely when $1 = I$, that is, when for each element a we have $a = 1a$.

We end the section with another illustration of the usefulness of the focus on the restriction functors. One says that a lambda algebra is a *lambda model* if the so-called Meyer-Scott axiom holds:

$$\text{if } ax = bx \text{ for all } x : 1 \rightarrow A \text{ then } 1a = 1b.$$

Proposition 5.14. *In the situation of the previous theorem, the following statements are equivalent:*

1. A is a lambda model
2. the category $C(A)$ has enough points
3. the functor $K(Q)$ is an equivalence of categories

Proof. We only prove the equivalence between the first and the third condition, referring to [Bar84] or [Koy84] for detailed proofs. The functor $K(Q)$ is always full on arrows and essentially surjective, regardless of whether A is a lambda model or not. If A is a lambda model, then consider two maps in $C(A)$, given by elements a, b . If these are sent by $K(Q)$ to the same morphism, that means that they represent the same function, i.e. that $ax = bx$ for all elements x . Now the Meyer-Scott axiom implies that $1a = 1b$, but by assumption on the elements from which we built $M(A)$ and hence $C(A)$, this implies $a = b$, so that $K(Q)$ is faithful. Conversely, if $K(Q)$ is faithful, then any two elements of $M(A)$ representing the same function must be equal. □

As a consequence, we get the characterization of extensionality:

Corollary 5.15. *In the situation of the previous theorem, the following statements are equivalent:*

1. A is extensional (meaning that $ax = bx$ for all x implies $a = b$)
2. the map $1 : I \rightarrow [I \rightarrow I]$ is an isomorphism in $C(A)$ and the functor $K(Q)$ is an equivalence of categories

Proof. If A is extensional, then it is also weakly extensional, since $a = b$ implies $1a = 1b$. Moreover, extensionality implies that $1 = I$, because $1xy = Ixy$ for all x, y (using extensionality twice). Conversely, if $1 = I$ and A is weakly extensional, then $ax = bx$ for all x implies $1a = 1b$, whence $Ia = Ib$, and therefore $a = b$, so that A is extensional. □

5.4 Notes and references

For detailed expositions of the original Scott-Koymans Theorem see Koymans' thesis [Koy84] or Barendregt's [Bar84]. In these texts, one also finds more on the various ways of presenting and characterizing lambda models.

Even and Pino Pérez were the first to extend the theorem to a partial setting; they announce the result in an abstract [Eve95], but it seems that a complete proof has not been published.

It should be noted that the analysis given here is both more detailed and more informative, even when restricted to the total case; as far as we aware, it has never been explained in the literature that a reflexive object already *is* a lambda algebra, and that the fact that its set of points is a lambda algebra is a consequence of this. Also, the use of the comparison functor $K(Q)$ seems to be new, as well as the fact that this functor being an equivalence amounts to having a lambda model.

We have not explicitly mentioned C-monoids; essentially, these are one-object ccc's, and are discussed in [Lam86]. Their role in the Scott-Koymans theorem is worked out in detail in [Koy84]. The restriction monoid $M(A)$ that we constructed from a PCA A could be called a restriction C-monoid, and one could work out a theory of such objects and their relation to partial lambda algebras, just as this has been done in the total case.

6 Examples

It is now time for some examples of partial lambda algebras. We present two (classes of) examples here, both living in the category of directed complete posets. The first is a modification of the D_∞ models. The second is a partial variation on Engeler's Graph Algebras.

6.1 Directed Complete Partial Orders

Let us begin by describing the restriction category in which we will work. We will outline all constructions, leaving the details to the reader.

Let $A = (A, \leq)$ be a poset. We say that A is *directed complete* if for each nonempty, directed subset $A' \subseteq A$ there exists a supremum $\bigvee A' \in A$. (Recall that a subset A' is called directed if for every pair $a, b \in A'$ there exists an element $c \in A'$ with $a \leq c, b \leq c$.) Typically, one also insists on having a supremum for the empty set, which is the same as a bottom element, but we do not impose this condition here. From now on, when we say "directed subset", we will mean "nonempty directed subset", and similarly for derived notions.

Every such directed complete poset A (abbreviated DCPO) comes equipped with the Scott-topology; a subset $X \subseteq A$ is open if and only if it is upwards closed and has the property that for each directed $A' \subseteq A$, if $\bigvee A' \in X$, then $A' \subseteq X \neq \emptyset$.

The category DCPO has as objects the DCPO's, and as morphisms partial continuous functions with open domain. Concretely, a partial function $f : A \rightarrow B$ is a morphism if f is orderpreserving and if for each directed A' with $\bigvee A' \in \text{dom}(f)$, we have

$$f(\bigvee A') = \bigvee (A' \cap \text{dom}(f)).$$

This is a restriction category, where the restriction of such f is given by the open inclusion $\text{dom}(f) \hookrightarrow A$.

The category DCPO is cartesian: both products and the terminal object are constructed as in the category of posets (caveat: the Scott-topology on the product need not coincide with the product topology).

Given two DCPOs A and B , their exponential $[A \multimap B]$ can be formed, by taking as underlying set the set $\text{DCPO}(A, B)$, that is, the set of all partial continuous functions with open domain from A to B . This set has a partial ordering via

$$f \leq g \Leftrightarrow \forall x \in \text{dom}(f) : x \in \text{dom}(g) \text{ and } f(x) \leq g(x).$$

Suprema of directed subsets are computed pointwise. More precisely, given a directed family $f_i : A \rightarrow B$, the supremum $\bigvee f_i$ is defined at $a \in A$ when there exists an index i for which $f_i(a)$ is defined. In this case, we put

$$(\bigvee f_i)(a) = \bigvee \{f_i(a) \mid f_i(a) \downarrow\}.$$

This indicates that DCPO is a cartesian closed restriction category; details are omitted.

6.2 The D_∞ -construction

We can now start looking for reflexive objects in the restriction category DCPO. Classically (working in DCPO's with total maps, and hence with ordinary exponentials) this problem was solved by Dana Scott; in fact, he constructed a lattice D_∞ with the property that $D_\infty \cong D_\infty^{D_\infty}$. We follow his construction, but replace the total function spaces in the construction by partial function spaces.

We start with an arbitrary DCPO D , which has a bottom element \perp . We then define inductively:

$$D_0 = D, \quad D_{n+1} = [D_n \multimap D_n].$$

There are canonical maps $\phi_0 : D_0 \rightarrow D_1$ and $\psi_0 : D_1 \rightarrow D_0$, given by $\phi_0(d) = \lambda x.d$ (the constant function with value d) and $\psi_0(f) = f(\perp)$ (evaluation at the bottom element). Note that ϕ_0 is total, but ψ_0 is not. These induce $\phi_n : D_n \rightarrow D_{n+1}$ and $\psi_n : D_{n+1} \rightarrow D_n$, given by

$$\phi_n(x) = \phi_{n-1} \circ x \circ \psi_{n-1}; \quad \psi_n(y) = \psi_{n-1} \circ y \circ \phi_{n-1}.$$

By induction on n , one shows that ϕ_n and ψ_n are continuous maps and that they satisfy the relations $\psi_n \circ \phi_n = 1$ and $\phi_n \circ \psi_n \leq 1$. Observe also that for $n > 0$, all ϕ_n and ψ_n are total maps.

We have a diagram

$$D_0 \xrightarrow{\phi_0} D_1 \xrightarrow{\phi_1} D_2 \xrightarrow{\phi_2} \dots$$

and we can form the colimit of this diagram. The vertex will be called D_∞ , and has the following explicit description.

An element of D_∞ is a partial function $\alpha : \mathbb{N} \rightarrow \bigcup_{n \in \mathbb{N}} D_n$ such that

- $\alpha(n) \downarrow \Rightarrow \alpha(n+1) \downarrow$ and $\alpha(n) = \psi_n(\alpha(n+1))$;
- $\alpha(1) \downarrow$.

Two such elements α, β of D_∞ are ordered via:

$$\alpha \leq \beta \Leftrightarrow \forall n \in \text{dom}(\alpha) : \beta(n) \downarrow \text{ and } \alpha(n) \leq \beta(n).$$

Suprema of directed subsets of D_∞ are computed pointwise, in the sense that

$$(\bigvee \{\alpha_i \mid i \in I\})(n) = \bigvee \{\alpha_i(n) \mid i \in I\}$$

where the left-hand side is defined whenever there exists $i \in I$ for which $\alpha_i(n)$ is defined. We leave as an exercise the verification that this indeed has the required properties.

A remark is in order here: we computed the *colimit* of the ascending chain $D_0 \xrightarrow{\phi_0} D_1 \xrightarrow{\phi_1} \dots$, but formulated this in terms of the maps ψ_n . In the

total case (meaning: in the category of DPCOs with total maps), one has limit-colimit coincidence here: the limit of the diagram $D_0 \xleftarrow{\psi_0} D_1 \xleftarrow{\psi_1} \dots$ is isomorphic to the colimit of the chain $D_0 \xrightarrow{\phi_0} D_1 \xrightarrow{\phi_1} \dots$. However, this breaks down in the partial setting. There are two reasons: first, the fact that the map ψ_0 is not total, and second, the fact that in a restriction category limits and colimits behave quite differently.

Without verifying the details, we state the main result:

Proposition 6.1. *The object D_∞ is a reflexive object in the cartesian closed restriction category DCPO. In fact, $D_\infty \cong [D_\infty \multimap D_\infty]$, so that D_∞ is an extensional lambda model.*

The proof follows the proof for the total case, see the references below. The claim about extensionality follows from the observation that the category DCPO has enough points.

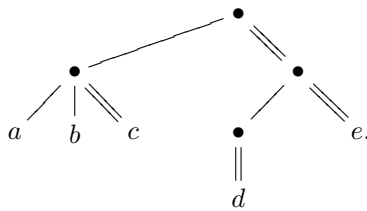
Note that nothing prevents us from starting with $D = D_0 = \{\perp\}$, the trivial DCPO with bottom element. In the total case, we would get $D_n = \{\perp\}$ for all n , but now that we work with partial function spaces we get a nontrivial object D_∞ . In a certain way, this is the smallest nontrivial solution to the problem of finding a reflexive object in DCPO.

6.3 Graph algebras and trees

Our second example also resides in the restriction category DCPO. It is essentially a reformulation of Engeler's construction of a graph algebra from a set A , which then has been made partial.

We start by fixing a set A , which will serve as a set of atoms. By a **tree** with leaves labelled in A , we mean a finite planar tree t together with an assignment $Leaf(t) \rightarrow A$, which associates to each leaf of t and element of A . The collection of such trees with leaves in A is denoted by $\mathcal{T}(A)$.

A typical finite tree over A looks like:



We always draw the rightmost outgoing edges double, since we wish to think of the distinguished edge as *output* and of the others as *input*. Therefore, a tree can be thought of as an *instruction*. The inputs and output of such an instruction are subtrees, and therefore are instructions themselves. If a node has only one outgoing edge (such as the one ending in d in the example), then this node is an instruction which doesn't need any input to produce its output.

If a node has no outgoing edges at all (i.e. is a leaf), then it cannot produce any output. Note also that the order of the input tree matters, as well as their multiplicity. A tree t with input branches t_1, \dots, t_n and output r will sometimes be written $t = \langle t_1, \dots, t_n; r \rangle$.

Definition 6.2. *A forest of trees labelled in A is a nonempty set of trees. We write $\mathcal{F}(A) = \mathcal{P}_{\neq \emptyset}(\mathcal{T}(A))$ for the collection of such forests.*

In order to avoid confusion, we stress that $\mathcal{F}(A)$ consists of (possibly infinite) forests, of which the trees are finite. The set of forests $\mathcal{F}(A)$ is partially ordered by inclusion and forms a DCPO when we take the supremum operation to be the union. As a DCPO, it is algebraic, meaning that every element is a supremum of compact (finite) elements. As a consequence, a continuous map on $\mathcal{F}(A)$ is determined by what it does on finite forests.

We will show that $\mathcal{F}(A)$ is in fact a reflexive object. First, there is a partial application on $\mathcal{F}(A)$: given two forests Φ, Ψ , define

$$\Phi \bullet \Psi = \{r \in \mathcal{T}(A) \mid \exists \langle t_1, \dots, t_n; r \rangle \in \Phi, t_1, \dots, t_n \in \Psi\},$$

where the left-hand side is defined whenever the set on the right-hand side is nonempty. Note as an aside that if we would have included the empty forest, then the above definition of application would still work and be total, by replacing “undefined” by the empty forest. Since this is an exposition about partial structures, however, we will not look further into this and get back to the partial setting.

Every forest Φ gives rise to a partial function $\Phi \bullet -$, which is easily seen to be determined by its action on finite forests and is therefore partial continuous. By the definition of definedness, it is also immediate that it has open domain. Explicitly, the domain of $\Phi \bullet -$ is the upwards closure of the collection of finite forests of the form $\{t_1, \dots, t_n\}$ for $\langle t_1, \dots, t_n; r \rangle \in \Phi$.

On the other hand, given a partial continuous function $f : \mathcal{F}(A) \rightarrow \mathcal{F}(A)$, we can construct a forest Φ_f by defining

$$\Phi_f = \{\langle t_1, \dots, t_n; r \rangle \mid r \in f(\{t_1, \dots, t_n\})\}.$$

Then Φ_f represents the function f , in the sense that for each forest Ψ , we have $f(\Psi) = \Phi_f \bullet \Psi$.

This defines maps $\mathcal{F}(A) \rightarrow [\mathcal{F}(A) \rightarrow \mathcal{F}(A)]$ and $[\mathcal{F}(A) \rightarrow \mathcal{F}(A)] \rightarrow \mathcal{F}(A)$, which are continuous (exercise). Therefore, $\mathcal{F}(A)$ is a reflexive object.

As an illustration of how certain combinators can be constructed in this model, we look at the combinator \mathbf{k} . One example of a forest playing the role of this combinator may be taken to be the set

$$\mathbf{k} = \left\{ \begin{array}{c} \bullet \\ \parallel \\ t \end{array} \quad \begin{array}{c} \bullet \\ \diagdown \\ t \end{array} \mid t \in \mathcal{T}(A) \right\}.$$

This is by no means the only possible choice for \mathbf{k} ; in fact, if one would unpack the meaning of $\llbracket \lambda xy.x \rrbracket$ in the model, one would end up with a larger set of trees.

6.4 Notes and references

The partial map category of DPCOs and partial continuous maps has been studied by Plotkin (*unpublished work) as a framework for domain theory. There is an alternative way of viewing this category, namely by considering DCPOs with a bottom element, and bottom-preserving continuous maps between them. The bottom element then represents “undefined”.

The construction of the D_∞ model in the partial setting is most likely known to several insiders, but, as far as we know, has not been published. For a detailed account of the total case, see either Scott’s paper [Sco72] or Barendregt’s exposition [Bar84].

Engeler’s graph algebra (see [Eng81]) was intended as a code-free version of Scott’s graph model $\mathcal{P}\omega$, and the tree model we presented here is a slight modification of the graph algebra. The relation is as follows. Where we deal with trees $\langle t_1, \dots, t_n; r \rangle$, Engeler deals with pairs $(\{t_1, \dots, t_n\}, r)$. Therefore, we have introduced some extra structure on our objects. As a consequence, there are several forests representing the same element of the graph algebra. In addition, we have omitted the empty forest and made the application partial.

The tree model allows for some curious variations: for example, one can drop the requirement that trees be finite, and replace it by the requirement that trees be finitely branching at each node. A discussion of how such models behave will be presented in [Hof05].

7 Appendix

In this appendix, we formulate the axioms A_β^R and prove that when these axioms are added to CL^R , the resulting theory is equivalent to the partial lambda calculus. The axiom set will be finite, and the axioms will be closed and total.

We will try to follow the proofs for the total case as given in [Bar84] as closely as possible, but the reader is warned that quite a few extensions and adaptations will have to be made. The reason for this is partly that the old axioms A_β have to be adapted in order to make sense in the partial setting, but also that we have to take care of all the axioms from CL^R that deal with the restriction.

First, we present a set of axiom *schemes*, which serve as an intermediate step. We show that when these axiom schemes are added to CL^R , the λ^* -operator in the resulting theory is sufficiently well-behaved (especially with respect to substitutions) and that the ξ -rule is valid. These axiom schemes, however, are not closed and are not finite. The second step, therefore, consists of showing that all these schemes follow from the finite set of closed axioms A_β^R , which are essentially suitable abstractions of the axiom schemes.

The axioms of A_β^R are not very intuitive when spelled out in terms of the combinators \mathbf{k} and \mathbf{s} ; for this reason, we introduce some shorthand notation, which, hopefully, provides the reader with a better grasp of the thrust of the axioms.

It will be easy to show that the axioms A_β^R are valid when translated into the lambda calculus; moreover, we can prove that the translation from the lambda calculus to CL^R becomes sound when we add these axioms. Then we have our main result (Theorem 3.13), which we repeat here for convenience:

Theorem 7.1. *The theories $CL^R + A_\beta^R$ and λ^R are equivalent (via the translations $(-)_{CL}$ and $(-)_\lambda$), in the sense that:*

1. $\lambda^R \vdash M = N \iff M_{CL,\lambda} = N_{CL,\lambda}$
2. $CL^R + A_\beta^R \vdash M = N \iff M_{\lambda,CL} = N_{\lambda,CL}$
3. $\lambda^R \vdash M = N \iff CL^R + A_\beta^R \vdash M_{CL} = N_{CL}$
4. $CL^R + A_\beta^R \vdash M = N \iff \lambda^R \vdash M_\lambda = N_\lambda$

In Table 3, we have listed the auxiliary axiom schemes. The first two axioms (which are already closed), are precisely what is needed to achieve the second item of the theorem:

Lemma 7.2. *For any term P we have:*

$$CL^R + P_1 + P_2 \vdash M = M_{\lambda,CL}.$$

Proof. Induction on the structure of M . The case where M is a variable is trivial. If M is \mathbf{k} or \mathbf{s} then we use P_1 or P_2 . Both translations $(-)_\lambda$ and $(-)_{CL}$ commute with application and restriction, so that the inductive steps are clear. \square

P_1	$\mathbf{k} = \lambda^*xy.x$	P_7	$\lambda^*x.x _{kx} = \lambda^*x.x$
P_2	$\mathbf{s} = \lambda^*xyz.xz(yz)$	P_8	$\lambda^*x.M _x = \lambda^*x.M$
P_3	$\mathbf{s}(\mathbf{k}P)(\mathbf{k}Q) _{PQ} = \mathbf{k}(PQ)$	P_9	$\lambda^*x.M _M = \lambda^*x.M$
P_4	$\lambda^*x.\mathbf{k}PQ = \lambda^*x.(P _Q)$	P_{10}	$\lambda^*x.M _{(N _D)} = \lambda^*x.M _{N,D}$
P_5	$\lambda^*x.\mathbf{s}PQR = \lambda^*x.PR(QR)$	P_{11}	$\lambda^*x.M _D N _E = \lambda^*x.(MN) _{D,E}$
P_6	$\lambda^*x.M _{sPQ} = \lambda^*x.M _{P,Q}$		

Table 3: Auxiliary schemes

Axiom scheme P_3 gives us the following result about the λ^* operator. Recall that in the total case, lambda abstraction can be simplified by defining $\lambda^+x.M$ as $\lambda^*x.M$, but by adding the clause $\lambda^+x.P = kP$ when $x \notin FV(P)$. This is simpler, since this allows one to skip inductive steps when x does not occur in P . In the total case one prefers λ^+ over λ^* because it behaves better with respect to substitution. However, using λ^+ in the partial case would lead to the undesirable consequence that $\lambda^+.P$ is not necessarily total anymore. Now axiom P_3 relates both translations.

Lemma 7.3. *Using P_3 , we have:*

$$CL + P_3 \vdash \lambda^+x.P = (\lambda^*x.P)|_D$$

where D is the set of subterms of P not containing x .

Proof. Induction on P , the crucial case being $P = MN$. Assume that x occurs in MN . Then:

$$\begin{aligned} \lambda^+x.MN &= \mathbf{s}(\lambda^+x.M)(\lambda^+x.N) \\ &= \mathbf{s}(\lambda^*x.M)(\lambda^*x.N)|_D \quad \text{by IH} \\ &= (\lambda^*x.MN)|_D \end{aligned}$$

where D is the set of subterms of MN not containing x . □

In particular, when the variable x is not free in P , we have that $(\lambda^*x.P)|_P = kP$.

The main point here, and thus one of the reasons for considering axiom scheme P_3 , is the following substitution lemma.

Lemma 7.4. *Let P be a CL^R -term, x, y distinct variables, and M a term not containing x . Then*

$$CL^R + P_3 \vdash (\lambda^*x.P[y := M])|_M = (\lambda^*x.P)[y := M]|_M.$$

Proof. Induction on P . If $P = x$, or P is a constant, then the statement is true even without restricting to M . If $P = y$, then the left hand side becomes

$(\lambda^*x.M)_{|M}$ and the right hand side becomes $(\mathbf{k}M)_{|M}$, which are equal, using the previous lemma. When $P = AB$, the calculation goes as follows:

$$\begin{aligned}
(\lambda^*x.AB[y := M])_{|M} &= (\lambda^*x.A[y := M]B[y := M])_{|M} \\
&= \mathbf{s}(\lambda^*x.A[y := M])(\lambda^*x.B[y := M])_{|M} \\
&= \mathbf{s}(\lambda^*x.A[y := M])_{|M}(\lambda^*x.B[y := M])_{|M} \\
&= \mathbf{s}((\lambda^*x.A)[y := M])(\lambda^*x.B)[y := M]_{|M} \quad \text{by IH} \\
&= \mathbf{s}(\lambda^*x.A)(\lambda^*x.B)[y := M]_{|M} \\
&= (\lambda^*x.AB)[y := M]_{|M}.
\end{aligned}$$

The case $P = A_{|D}$ is similar. \square

The function of the schemes P_3, \dots, P_8 is to make sure that the operation λ^* in CL^R satisfies the following technical property.

Lemma 7.5. *For any term M with u not occurring free in M , we have*

$$CL^R + P_3 + P_4 + P_5 + P_6 + P_7 + P_8 \vdash \lambda^*u.(\lambda^*x.M)u = \lambda^*x.M.$$

Proof. This is induction on M .

Case $M = x$:

$$\begin{aligned}
\lambda^*u.(\lambda^*x.M)u &= \lambda^*u.(\mathbf{s}\mathbf{k}\mathbf{k})u \\
&= \lambda^*u.\mathbf{k}u(\mathbf{k}u) \quad \text{by } P_5 \\
&= \lambda^*u.u_{|\mathbf{k}u} \quad \text{by } P_4 \\
&= \lambda^*u.u \quad \text{by } P_6 \\
&\equiv \lambda^*x.x
\end{aligned}$$

Case $M = c$, where c is either a constant or a variable different from x :

$$\begin{aligned}
\lambda^*u.(\lambda^*x.M)u &= \lambda^*u.(\mathbf{k}c)u \\
&= \lambda^*u.c_{|u} \\
&= \lambda^*u.c.
\end{aligned}$$

The last step follows from axiom P_8 using P_3 : applying both sides to c gives $\lambda^*u.c_{|u} =_{def} \mathbf{s}(\mathbf{s}(\mathbf{k}\mathbf{k})(\lambda^*u.c))(\lambda^*u.u) = \mathbf{s}(\mathbf{s}(\mathbf{k}\mathbf{k})(\mathbf{k}c))(\mathbf{s}\mathbf{k}\mathbf{k}) = \mathbf{s}(\mathbf{s}(\mathbf{k}c))(\mathbf{s}\mathbf{k}\mathbf{k}) = \mathbf{k}c = \lambda^*u.c$.

Case $M = PQ$:

$$\begin{aligned}
\lambda^*u.(\lambda^*x.M)u &= \lambda^*u.s(\lambda^*x.P)(\lambda^*x.Q)u \\
&= \lambda^*u.(\lambda^*x.P)u((\lambda^*x.Q)u) \\
&= \mathbf{s}(\lambda^*u.((\lambda^*x.P)u))(\lambda^*u.((\lambda^*x.Q)u)) \\
&= \mathbf{s}(\lambda^*x.P)(\lambda^*x.Q) \quad \text{by IH} \\
&= \lambda^*x.PQ.
\end{aligned}$$

Case $M = P|_Q$:

$$\begin{aligned}
\lambda^*u.(\lambda^*x.P|_Q)u &= \lambda^*u.(\mathbf{s}(\mathbf{s}(\mathbf{k}\mathbf{k})(\lambda^*x.P))(\lambda^*x.Q))u \text{ by definition of } \lambda^*x \\
&= \lambda^*u.(\mathbf{s}(\mathbf{k}\mathbf{k})(\lambda^*x.P))u((\lambda^*x.Q)u) \text{ using } P_5 \\
&= \mathbf{s}(\lambda^*u.\mathbf{s}(\mathbf{k}\mathbf{k})(\lambda^*x.P)u)(\lambda^*u.(\lambda^*x.Q)u) \text{ by definition of } \lambda^*u \\
&= \mathbf{s}(\lambda^*u.\mathbf{k}\mathbf{k}u((\lambda^*x.P)u))((\lambda^*u.(\lambda^*x.Q)u)) \text{ using } P_5 \\
&= \mathbf{s}(\lambda^*u.\mathbf{k}((\lambda^*x.P)u))((\lambda^*u.Q)u) \text{ using } P_4 \\
&= \mathbf{s}(\mathbf{s}(\lambda^*u.\mathbf{k})(\lambda^*u.(\lambda^*x.P)u))(\lambda^*u.(\lambda^*x.Q)u) \text{ definition of } \lambda^*u \\
&= \mathbf{s}(\mathbf{s}(\lambda^*u.\mathbf{k})(\lambda^*x.P))(\lambda^*x.Q) \text{ by IH} \\
&= \mathbf{s}(\mathbf{s}(\mathbf{k}\mathbf{k})(\lambda^*x.P))(\lambda^*x.Q) \\
&= \lambda^*x.P|_Q
\end{aligned}$$

□

In fact, we can prove the stronger

$$CL^R + P_3 + \dots + P_8 \vdash \lambda^*v.[(\lambda^*x.M)v]_{|(\lambda^*x.N)v} = \lambda^*x.M|_N, \quad v \text{ not free in } M, N.$$

This is true, since

$$\begin{aligned}
\lambda^*v.[(\lambda^*x.M)v]_{|(\lambda^*x.N)v} &= \mathbf{s}(\mathbf{s}(\mathbf{k}\mathbf{k})(\lambda^*v.(\lambda^*x.M)v))(\lambda^*v.(\lambda^*x.N)v) \\
&= \mathbf{s}(\mathbf{s}(\mathbf{k}\mathbf{k})(\lambda^*x.M))(\lambda^*x.N) \\
&= \lambda^*x.M|_N.
\end{aligned}$$

The second step is an application of the lemma.

Next, we turn attention to the ξ -rule. First, let us agree to call an equation $M = N$ *total* if both M and N are total, i.e. $P|_M = P = P|_N$ for all P . We wish to show that for any set \mathcal{S} of total, closed equations, if $CL^R + \mathcal{S} \vdash P_3, \dots, P_{11}$, then $CL^R + \mathcal{S}$ is closed under the ξ -rule.

The axioms of CL^R governing equality are trivial. For the rule $M = N \implies MZ = NZ$, we observe that $\lambda^*x.(MZ) = \mathbf{s}(\lambda^*x.M)(\lambda^*x.Z) =_{IH} \mathbf{s}(\lambda^*x.N)(\lambda^*x.Z) = \lambda^*x.(NZ)$, and similarly for the dual rule.

For every other axiom of CL^R , there is a corresponding scheme stating that applying λ^* on both sides preserves validity.

Finally, if we have an axiom $M = N$ from \mathcal{S} , then we have $\lambda^*x.M = (\lambda^*x.M)|_M = kM = kN = (\lambda^*x.N)|_N = \lambda^*x.N$, using that $x \notin FV(M, N)$ and that M, N are total.

We now solve the problem of replacing the given schemes by closed axioms. One of the problems here is, that the axioms would become very large when spelled out in terms of \mathbf{k} and \mathbf{s} only, to the extent that they would be incomprehensible. Therefore we introduce some shorthand notation, which will help us to state the axioms in a convenient way, as well as show that they satisfy the right properties.

Definition 7.6. Let M be a term, and let u, \vec{x} be variables, where u is not equal to any of the \vec{x} . By induction on the structure of M , we define a term $\Theta_u(M; \vec{x})$:

- $\Theta_u(x; \vec{x}) = 1x$ if x is one of the \vec{x}
- $\Theta_u(P; \vec{x}) = \lambda^*u.P$ if none of the \vec{x} occur in P
- $\Theta_u(PQ; \vec{x}) = \mathbf{s}(\Theta_u(P; \vec{x}))(\Theta_u(Q; \vec{x}))$
- $\Theta_u(P|_Q; \vec{x}) = \mathbf{s}(\mathbf{s}(\mathbf{kk})\Theta_u(P; \vec{x}))(\Theta_u(Q; \vec{x}))$

Finally, we put

$$\Lambda_u(M; \vec{x}) = \lambda^*\vec{x}.\Theta_u(M; \vec{x}).$$

The idea is, that $\Theta_u(M; \vec{x})$ is like $\lambda^*u.M$, but with the occurrences of $\lambda^*u.x$ replaced by $1x$. The point of the definition is the following lemma.

Lemma 7.7. Let M be a term, \vec{x}, u variables with u distinct from the \vec{x} , and let \vec{N} be arbitrary terms fitting in \vec{x} . Then

$$CL^R + P_3 + \dots + P_8 \vdash [\Lambda_u(M; \vec{x})](\lambda^*u.N_1) \dots (\lambda^*u.N_k) = \lambda^*u.M[\vec{x} := \vec{N}].$$

Proof. We first observe that $1(\lambda^*u.M) = \lambda^*u.M$, using Lemma 7.5. Now we do an induction on the structure of M . If $M = x_i$, where x is one of the \vec{x} , then we obtain $(\lambda^*\vec{x}.1x)(\lambda^*u.N_1) \dots (\lambda^*u.N_n) = 1(\lambda^*u.N_i) = \lambda^*u.N_i$, as required.

If M is \vec{x} -free, then $(\lambda^*\vec{x}.\lambda^*u.M)(\lambda^*u.N_1) \dots (\lambda^*u.N_n) = \lambda^*u.M$.

If $M = PQ$, then

$$\begin{aligned} \Lambda_u(PQ; \vec{x})(\lambda^*u.\vec{N}) &= [\lambda^*\vec{x}.\mathbf{s}(\Theta_u(P; \vec{x}))(\Theta_u(Q; \vec{x}))](\lambda^*u.\vec{N}) \\ &= \mathbf{s}(\Theta_u(P; \vec{x})[\vec{x} := \lambda^*u.\vec{N}])(\Theta_u(Q; \vec{x})[\vec{x} := \lambda^*u.\vec{N}]) \\ &= \mathbf{s}(\lambda^*u.P[\vec{x} := \vec{N}])(\lambda^*u.Q[\vec{x} := \vec{N}]) \quad \text{by IH} \\ &= \lambda^*u.PQ[\vec{x} := \vec{N}]. \end{aligned}$$

The case $M = P|_Q$ is similar. □

Note that if all of the free variables of M are among u, \vec{x} , the term $\Lambda_u(M; \vec{x})$ is closed and total.

We are now ready to state the axiom set A_β^R , which is displayed in table 4. The first five of these axioms are almost the same as for the total case; the only difference is that in the third and the fifth axiom we have to add a suitable restriction. The other axioms are those which will generate the schemes which we used to get the validity of the ξ -rule.

We first show that all the auxiliary schemes follow from the axioms A_β^R . Scheme P_3 follows by applying axiom A.3 to P and Q . Scheme P_5 follows from A.5 by applying both sides to $\lambda^*x.P, \lambda^*x.Q$ and $\lambda^*x.R$.

Scheme P_6 is obtained by applying A.9 to $\lambda^*u.P, \lambda^*u.Q$ and $\lambda^*u.R$, using Lemma 7.7. Schemes $P_7 - P_{11}$ follow in a similar fashion.

A.1	$\mathbf{k} = \lambda^*xy.\mathbf{k}xy$
A.2	$\mathbf{s} = \lambda^*xyz.\mathbf{s}xyz$
A.3	$\lambda^*xy.\mathbf{s}(\mathbf{k}x)(\mathbf{k}y) _{xy} = \lambda^*xy.\mathbf{k}(xy)$
A.4	$\lambda^*xy.\mathbf{s}(\mathbf{s}(\mathbf{k}\mathbf{k})x)y = \lambda^*xyz.xz _{yz}$
A.5	$\lambda^*xyz.\mathbf{s}(\mathbf{s}(\mathbf{s}(\mathbf{k}\mathbf{s})x)y)z = \lambda^*xyz.\mathbf{s}(\mathbf{s}xz)(\mathbf{s}yz)$
A.6	$\Lambda_u(x _x; x) = \Lambda_u(x; x)$
A.7	$\Lambda_u(x _u; x) = \Lambda_u(x; x)$
A.8	$\Lambda_u(x _{kx}; x) = \Lambda_u(x; x)$
A.9	$\Lambda_u(x _{syx}; x, y, z) = \Lambda_u(x _{y,z})$
A.10	$\Lambda_u(x _y u _v; x, y, u, v) = \Lambda_u((xu) _{yv}; x, y, u, v)$
A.11	$\Lambda_u(x _{y z}; x, y, z) = \Lambda_u(x _{y,z}; x, y, z)$

Table 4: Axioms A_β^R

Now P_1 and P_2 are immediate from A.1 and A.2, and it remains to be proved that P_4 follows. Compute:

$$\begin{aligned}
\lambda^*x.\mathbf{k}PQ &= \mathbf{s}(\lambda^*x.\mathbf{k}P)(\lambda^*x.Q) \\
&= \mathbf{s}(\mathbf{s}(\mathbf{k}\mathbf{k})(\lambda^*x.P))(\lambda^*x.Q) \\
&= (\lambda^*xy.\mathbf{s}(\mathbf{s}(\mathbf{k}\mathbf{k})x)y)(\lambda^*x.P)(\lambda^*x.Q) \\
&= (\lambda^*xyz.xz|_{yz})(\lambda^*x.P)(\lambda^*x.Q) \quad \text{by A.4} \\
&= \lambda^*z.(\lambda^*x.P)z|_{(\lambda^*x.Q)z} \\
&= \lambda^*x.P|_Q \quad \text{by the substitution lemma.}
\end{aligned}$$

Thus we have shown:

Proposition 7.8. *The axiom schemes P_1 - P_{11} are derivable from the axioms A_β^R , i.e.*

$$CL^R + A_\beta^R \vdash P_1, \dots, P_{11}.$$

Corollary 7.9. *The theory $CL^R + A_\beta^R$ is closed under the ξ -rule.*

Next, we need:

Lemma 7.10. *For each axiom $M = N$ of A_β^R , $\lambda^R \vdash M_\lambda = N_\lambda$.*

Proof. The first five axioms are proved in the same way as in the total case. For the axioms involving the Λ_u -terms, observe that all axioms have the same parameter variables. Therefore it is enough to show that the translations of the corresponding Θ_u -terms give valid equations.

For example, the axiom A.6 is dealt with as follows. The left hand side

reduces to:

$$\begin{aligned}
(\Theta_u(x|_x; x))_\lambda &= (\mathbf{s}(\mathbf{s}(\mathbf{kk})(1x))(1x))_\lambda \\
&= \mathbf{S}(\mathbf{S}(\mathbf{KK})(\lambda z.xz))(\lambda z.xz) \\
&= \lambda w.\mathbf{S}(\mathbf{KK})(\lambda z.xz)w((\lambda z.xz)w) \\
&= \lambda w.\mathbf{KK}w((\lambda z.xz)w)((\lambda z.xz)w) \\
&= \lambda w.\mathbf{KK}w(xw)(xw) \\
&= \lambda w.\mathbf{K}(xw)(xw) \\
&= \lambda w.xw|_{xw} \\
&= \lambda w.xw.
\end{aligned}$$

The last step uses the rule $M|_M = M$ (as well as the ξ -rule) in the λ^R -calculus. Similarly the right hand side reduces to

$$\begin{aligned}
(\Theta_u(x; x))_\lambda &= (1x)_\lambda \\
&= (\lambda^*u.xu)_\lambda \\
&= \lambda w.xw.
\end{aligned}$$

The other axioms are similar, only more elaborate. □

Since λ^R proves the translations of all the A_β^R -axioms, we now obtain:

$$CL^R + A_\beta^R \vdash M = N \implies \lambda^R \vdash M_\lambda = N_\lambda.$$

The remaining piece of the problem is to show that $(-)_CL$ is sound. In order to show this, we need to show that the translation is well-behaved with respect to substitution.

Lemma 7.11. *For terms M, N , we have*

$$CL^R + P_3 \vdash (M[y := N]|_N)_{CL} = M_{CL}[y := N_{CL}]|_{N_{CL}}.$$

Proof. Induction on the structure of M . Almost all cases are obvious (and hold even without restrictions), and we only show the case where $M = \lambda x.P$. There are two subcases, namely when $y \in FV(P)$ and when $y \notin FV(P)$. The latter case is easy, and we show only the former case here.

$$\begin{aligned}
&(\lambda^*x.(P[y := N]_{CL}))|_{N_{CL}} \\
&= (\lambda^*x.(P[y := N]|_N)_{CL})|_{N_{CL}} \quad \text{since } y \in FV(P) \\
&= (\lambda^*x.(P[y := N]_{CL})|_{N_{CL}})|_{N_{CL}} \quad \text{rule } \xi \\
&= (\lambda^*x.P_{CL}[y := N_{CL}]|_{N_{CL}})|_{N_{CL}} \quad \text{by IH} \\
&= (\lambda^*x.P_{CL}[y := N_{CL}])|_{N_{CL}} \quad \text{again since } y \in FV(P) \\
&= (\lambda^*x.P_{CL})[y := N_{CL}]|_{N_{CL}} \quad \text{by Lemma 7.4} \\
&= (\lambda x.P)_{CL}[y := N_{CL}]|_{N_{CL}}.
\end{aligned}$$

□

Now we are ready to put everything together and prove the main theorem:

Proof. (Of Theorem 3.13.) The first claim was already proved in Lemma 3.12. The second claim is Lemma 7.2. We have also established that the interpretation of $CL^R + A_\beta^R$ into λ^R is sound, so it remains to be seen that the interpretation of λ^R into $CL^R + A_\beta^R$ is sound, for then the remaining implications follow using the first two items.

So we have to prove that $\lambda^R \vdash M = N$ implies $CL^R + A_\beta^R \vdash M_{CL} = N_{CL}$, and we will do so by induction on the proof of $M = N$. Since the translation $M \mapsto M_{CL}$ by definition commutes with application and restriction, axioms 1-9 of λ^R are immediate.

Axiom 10, $M|_{\lambda x.N} = M$, translates into $(M_{CL})|_{\lambda^* x.N_{CL}} = M_{CL}$, which holds by Proposition 3.8, item (ii).

Since $CL^R + A_\beta^R \vdash P_1, \dots, P_{11}$ by Proposition 7.8, the theory $CL^R + A_\beta^R$ is closed under the ξ -rule by Corollary 7.9. The only axiom left to check is β -reduction. We compute:

$$\begin{aligned}
((\lambda x.M)N)_{CL} &= (\lambda x.M)_{CL} N_{CL} \\
&= (\lambda^* x.M_{CL})_{N_{CL}} \\
&= M_{CL}[x := N_{CL}]_{N_{CL}} && \text{by Proposition 3.8} \\
&= (M[x := N]_{CL})_{N_{CL}} \\
&= (M[x := N]_{N})_{CL}.
\end{aligned}$$

This completes the proof. □

References

- [Asp85] A. Asperti and G. Longo. Categories of partial morphisms and the relation between type-structures. Technical Report S-7-85, Dip. di Informatica, Univ. di Pisa, 1985.
- [Bar84] H.P. Barendregt. *The lambda calculus. Its syntax and semantics*. Revised edition. Studies in Logic and the foundations of mathematics 103, Elsevier, 1984.
- [Bee85] M.J. Beeson. *Foundations of constructive mathematics*. Springer Verlag, New York-Berlin, 1985.
- [Bet88] I. Bethke. Notes on partial combinatory algebras. Ph.D. Thesis, University of Amsterdam, 1988.
- [Coc02] J.R.B. Cockett and S. Lack. Restriction categories I: categories of partial maps. *Theoretical Computer Science* **270** (2002), 223–259.
- [Coc03] J.R.B. Cockett and S. Lack. Restriction categories II: partial map classification. *Theoretical Computer Science* **294** (2003), 61–102.
- [Coc04] J.R.B. Cockett and S. Lack. Restriction categories III: partial structures. Submitted for publication.
- [Cur87] P-L. Curien and A. Obtulowicz. Partiality, cartesian closedness and toposes. *Information and Computation* **80** (1989), 50–95.
- [Eng81] E. Engeler. Algebras and combinators. *Algebra Universalis* **13(3)** (1981), 389–392.
- [Eve95] C. Even and R. Pino Pérez. Extension of Scott-Koymans theorem to partial framework (extended abstract). *Bulletin of Symbolic Logic* **4(1)** (1995), 489–499.
- [Eve92] C. Even and R. Pino Pérez. λ_p -calculus and algebras with partial elements. XII International Conference of the Chilean Computer Science Society, Santiago, Chile, October 14-16th, 1992.
- [Hin80] R. Hindley and J. Seldin, eds. *To H.B. Curry: essays in combinatory logic, lambda calculus and formalisms*. Academic Press, 1980.
- [Hof03] P.J.W. Hofstra J. van Oosten. Ordered partial combinatory algebras. *Math. Proc. Cam. Phil. Soc.* **134** (2003) 445–463.
- [Hof05] P.J.W. Hofstra. A tree model for the lambda calculus. Unpublished Manuscript, 2006.
- [Koy84] C.P.J. Koymans. Models of the lambda calculus. Ph.D. Thesis, University of Amsterdam, 1984.

- [Lam86] J. Lambek and P.J. Scott. *Introduction to higher order categorical logic*. Cambridge studies in advanced mathematics 7, Cambridge University Press, 1986.
- [Lon94] J. Longley. Realizability toposes and language semantics. Ph. D. Thesis, University of Edinburgh, 1994.
- [Lon90] G. Longo and E. Moggi. A category-theoretic characterization of functional completeness. *Theoretical Computer Science* **70(2)** (1990), 193–211.
- [Mey82] A. Meyer. What is a model of the lambda calculus? *Information and Control* **52** (1982), 87–122.
- [Mog88] E. Moggi. The partial lambda-calculus. Ph.D. Thesis, University of Edinburgh, 1988.
- [Pao86] R. Di Paolo and A. Heller. Dominical categories. *Journal of symbolic logic*, 1986.
- [Plo75] G. Plotkin. Call-by-value, call-by-name and the λ -calculus. *Theoretical Computer Science* **1** (1975).
- [Rob86] E. Robinson and G. Rosolini. Categories of partial maps. *Information and Computation* **79** (1988), 95–130.
- [Sch04] L. Schröder. The logic of the partial λ -calculus with equality.
- [Sco72] D.S. Scott. Continuous lattices. In: F.W. Lawvere, ed. *Toposes, Algebraic Geometry and Logic*. Lecture Notes in Mathematics 274, Springer Verlag, 1972.
- [Sco79] D.S. Scott. Identity and existence in intuitionistic logic. In: Fourman, Mulvey, eds. *Applications of Sheaves*. Lecture notes in mathematics 753, Springer Verlag, 1979.
- [Tro73] A.S. Troelstra, ed. *Metamathematical investigations of intuitionistic arithmetic and analysis*. Lecture notes in computer science 344, Springer Verlag, 1973.